POLITECHNIKA ŚLĄSKA Biblioteka Główna Gliwice
P. 1877/92

4 1992

# informatyka

WYDAWNICTWO SIGMA NOT

Cena 18 000 zł

# informatyka

nr 4 1992
Kwiecień
Rok wydania XXVII

Miesięcznik
ISSN 0542-9951
INDEKS 36124



KOLEGIUM REDAKCYJNE:

mgr Jarosław DEMINET
mgr inż. Piotr FUGLEWICZ
mgr Teresa JABŁOŃSKA
(sekretarz redakcji)
Władysław KLEPACZ
(redaktor naczelny)
dr inż. Wojciech MOKRZYCKI
mgr inż. Jan RYŻKO
dr Zdzisław SZYJEWSKI
mgr Hanna WŁODARSKA – MARCZENKO

PRZEWODNICZĄCY RADY PROGRAMOWEJ:

Prof. dr hab.
Juliusz Lech KULIKOWSKI



WYDAWCA:
Wydawnictwo Czasopism i Książek Technicznych SIGMA NOT
Spółka z o.o.
ul. Biała 4
00-950 WARSZAWA
skrytka pocztowa 1004

Redakcja:
01-552 Warszawa,
Pl. Inwalidów 10, p. 104, 105
tel. 39-14-34

Materiałów nie zamówionych redakcja nie zwraca



**W sprawach ogłoszeń prosimy zwracać się bezpośrednio do Redakcji lub Działu Reklamy i Marketingu**
**00-950 Warszawa**
**ul. Biała 4**
**telefon: 20-31-24**
**telefaks: 20-31-16**
**teleks: 814550**


## W numerze:


| | Strona |
|---|---|
| Wykorzystanie sieci komputerowych w dydaktyce – *Kazimierz Kowalski, Andrzej Siemiński* | 1 |
| Graficzne stacje robocze SPARCstation – *Wiesław Przetacznik* | 5 |
| Komputerowe układy sterowania numerycznego – *Bogdan Kietzman* | 10 |
| Koncepcja symulacji obiektowej z wykorzystaniem języka Turbo Pascal 5.5 – *Krystyna Koleśnik* | 14 |
| Metody automatycznego rozpoznawania znaków – *Iwona Dąbrowska, Dorota Depowska* | 19 |
| Badania krajowego rynku komputerowego – *Maria Jerczyńska* | 23 |
| **Ze świata** | 25 |
| **/unix** | |
| W UNIX-ie ostrożność popłaca – *Piotr Carlson* | 27 |


## W najbliższych numerach:

- **D. Gaik, J. Kotula i J. Piecha omawiają ASYS-3 – system narzędzi graficznych opracowany w Zakładzie Elektroniki i Informatyki Uniwersytetu Śląskiego.**
- **Piotr Kowalski opisuje program T³ do przetwarzania tekstów technicznych, podaje główne zasady posługiwania się nim i jego elementy użytkowe.**
- **Dariusz Caban i Jacek Jarnicki zajmują się zagadnieniem filtrów obrazów cyfrowych.**
- **Bogdan Marzec naświetla problematykę standaryzacji systemów czasu rzeczywistego.**

## Warunki prenumeraty

**Przyjęcie prenumeraty** – wyłącznie na podstawie dokonanej wpłaty na drukach dostarczanych dotychczasowym prenumeratorom przez Wydawnictwo, lub nowym – po uprzednim zgłoszeniu zapotrzebowania (pisemnie lub telefonicznie) w Zakładzie Kolportażu Wydawnictwa.

**Blankiet wpłaty** – powinien zawierać następujące informacje: dokładna nazwa i adres (z kodem pocztowym) zamawiającego, tytuły zamawianych czasopism, ich liczbę i okres prenumeraty.

**Wpłata** – zgodnie z podanymi cenami należy dokonać w banku lub w UPT na konto podane na naszym blankiecie, tj:
**Państwowy Bank Kredytowy III O/Warszawa nr: 370015-1573-139-11**

**Prenumeratorzy zbiorowi** – osoby prawne obowiązują blankiety „Wpłata-Zamówienie". Cena normalna.

**Prenumeratorzy indywidualni** – osoby fizyczne obowiązują blankiety typu przekazy dla wpłat na rachunki bankowe. Cena normalna.

**Prenumerata ulgowa** – zgodnie z podaną ceną ulgową przysługuje wyłącznie osobom fizycznym, będącym członkami SNT, studentom i uczniom szkół zawodowych. Uczniowie szkół ogólnokształcących mogą zamówić w prenumeracie ulgowej tylko miesięcznik „Aura"

Uwaga! w podanym okresie prenumeraty można zamówić tylko po jednym egzemplarzu z każdego tytułu.

**Prenumerata ze zleceniem wysyłki** za granicę – cena prenumeraty ze zleceniem wysyłki za granicę jest dwukrotnie wyższa od ceny normalnej.

Należy podać dokładny adres odbiorcy za granicą.

**Terminy przyjmowania prenumeraty:**
– do 10 listopada na I, II, III, IV kwartał następnego roku
– do 28 lutego na II, III, IV kwartał br.
– do 31 maja na III i IV kwartał br.
– do 31 sierpnia na IV kwartał br.

Zmiany w prenumeracie, np. zmiana liczby tytułów, liczby egzemplarzy, rezygnacja z prenumeraty, można zgłaszać tylko w podanych terminach z mocą obowiązującą od następnego kwartału.

**Egzemplarze archiwalne** (z lat ubiegłych)
Można nabyć za gotówkę w Klubie Prasy Technicznej, Warszawa, ul. Mazowiecka 12 (tel. 26-80-16) lub zamówić pisemnie w Zakładzie Kolportażu, Dział Handlowy, 00-950 Warszawa, skr. poczt. 1004 (tel. 40-37-31), na rachunek lub za zaliczeniem pocztowym.

**Informacji o prenumeracie udziela:** Zakład Kolportażu Wydawnictwa SIGMA-NOT Spółka z o.o., 00-716 Warszawa, ul. Bartycka 20, skr. 1004. Telefony: 40-00-21 wewn. 293, 295, 299 lub 40-30-86, 40-35-89.

**Wstępna cena** jednego egzemplarza na 1992 rok: normalna – 18 000 zł, ulgowa – 13 500zł

**Wartość prenumeraty (w zł):**
**Normalna:** kwartalna – 54 000, półroczna 108 000, roczna 216 000
**Ulgowa:** kwartalna – 40 500, półroczna 81 000, roczna 162 000

**Uwaga: W przypadku zmiany cen w okresie objętym prenumeratą, prenumeratorzy zobowiązani są do dopłaty różnicy cen.**


Skład i druk: Drukarnia SIGMA NOT Sp. z o.o. z. 84/92

KAZIMIERZ KOWALSKI
California State University
ANDRZEJ SIEMIŃSKI
Politechnika Wrocławska

POLITECHNIKA ŚLĄSKA Biblioteka Główna
P. 1877/92

# Wykorzystanie sieci komputerowych w dydaktyce

**Przyłączenie uczelni polskich do sieci komputerowej EARN (*European Academic Research Network*), a poprzez nią – co innych sieci międzynarodowych, wzbudziło w środowisku akademickim duże zainteresowanie możliwościami wykorzystania takich sieci w wielu dziedzinach, w tym także w dydaktyce.**

**W tym artykule skupiono się na zastosowaniu sieci komputerowych w procesie dydaktycznym. W dziedzinie tej przodują uczelnie z USA. Wydaje się jednak, że zwiększająca się popularność sieci sprawi, iż będzie je można wykorzystać już niedługo przynajmniej w części uczelni polskich.**

Wprowadzenie komputerów do szkół wyższych przebiega wieloetapowo. W pierwszym etapie są one wykorzystywane na kierunkach informatycznych do nauki przedmiotów zawodowych, np. w trakcie zajęć laboratoryjnych z języków programowania. Duża dostępność bardzo dobrze oprogramowanych mikrokomputerów zgodnych z IBM PC/AT jest niewątpliwie główną przyczyną obserwowanego ostatnio na tych kierunkach podniesienia poziomu nauczania. Niejednokrotnie studenci dopiero rozpoczynający naukę np. jednego z języków programowania mają już umiejętności dorównujące umiejętnościom studentów kończących podobny kurs kilka lat temu. Trzeba jednak podkreślić, że biegłość w pisaniu instrukcji zwykle idzie w parze z bardzo poważnymi brakami w zakresie metodologii programowania i projektowania. Duża część zajęć tego typu może być z pełnym powodzeniem prowadzona na nie połączonych w sieć mikrokomputerach.

Drugi etap to wykorzystanie komputerów w procesie dydaktycznym na pozostałych kierunkach studiów przy nauczaniu przedmiotów nieinformatycznych. Etap ten ma potencjalnie jeszcze większe znaczenie. Obejmuje on środowiska nie związane profesjonalnie z informatyką i przygotowuje społeczeństwo do nadchodzącej rewolucji informatycznej. Możliwości wykorzystania komputerów na tym etapie są wielorakie. Dotyczą one nie tylko czynności bardzo prostych, jak np. opracowywanie sprawozdań laboratoryjnych [16], ale obejmują także posługiwanie się programami edukacyjnymi i symulacyjnymi, wyspecjalizowanymi systemami ekspertowymi [11] oraz umożliwiają wyszukiwanie informacji z baz danych zapisanych na dyskach optycznych. Stosowanie komputerów łączy się również z opanowaniem umiejętności posługiwania się dużą liczbą programów przydatnych w pracy zawodowej, takich jak np. procesory tekstu, arkusze kalkulacyjne lub programy DTP (ang. *Desk Top Publishing*) [5]. Pozwala to przypuszczać, że pokolenie tak wykształcone będzie świadome możliwości (ale i ograniczeń) stosowania komputerów i znajdzie lepsze wykorzystanie dla przykrytych obecnie pokrowcami mikrokomputerów. Podobnie, jak poprzednio, również w tym przypadku w zasadzie wystarczające są komputery pracujące samodzielnie, choć wykorzystuje się czasami sieci komputerowe [6, 12].

W tym artykule opisujemy możliwości, jakie uzyskuje użytkownik po przyłączeniu komputerów do sieci. Możliwe staje się wówczas porozumienie się osób za pomocą komputerów, tzw. komunikowanie się komputerowe (KK). Komputery stają się wówczas nowym środkiem przekazu. Korzyści uzyskane dzięki temu są wielorakie i wykraczają niejednokrotnie poza bezpośrednie korzyści utylitarne [4, 17]. Podstawową zaletą komunikowania się komputerowego jest zwiększenie możliwości współpracy między bardzo oddalonymi od siebie ludźmi, co może zwiększyć efektywność pracy przez wymianę pomysłów, idei i uwag. Przewiduje się, że możliwość odbywania konferencji za pomocą sieci może w bardzo znacznym stopniu wpłynąć zarówno na proces kształcenia, jak i prowadzenia badań naukowych w środowiskach akademickich. Dzięki sieci komputerowej użytkownik ma dostęp do ogromnej ilości danych i jest zmuszony do podejmowania szybko decyzji o tym, co należy przeczytać, a co pominąć. Nabranie takiej umiejętności może być ubocznym, ale istotnym efektem korzystania z sieci komputerowych.

## Komunikowanie się komputerowe

Po połączeniu komputerów w sieć, użytkownik uzyskuje bardzo wiele nowych możliwości. Będąc użytkownikiem jednego komputera może korzystać z takich zasobów innych komputerów, jak: wyspecjalizowane programy użytkowe, bazy danych, trudno dostępne urządzenia peryferyjne (np. bardzo

Dr inż. KAZIMIERZ KOWALSKI ukończył w 1970 r. Wydział Elektroniki Politechniki Wrocławskiej. W 1974 r. obronił doktorat na temat systemów wieloprocesorowych. W latach 1970–1974 zatrudniony był w Instytucie Cybernetyki Technicznej Politechniki Wrocławskiej, a następnie na Wydziale Informatyki i Zarządzania tej samej uczelni. Począwszy od roku 1986 pracuje na wydziale Copmputer Science w California State University, California, USA. Zainteresowania zawodowe koncentrują się na systemach wieloprocesorowych i sieciach komputerowych; jest administratorem universyteckiej sieci komputerowej. Dorobek naukowy obejmuje ponad 40 prac z zakresu sieci komputerowych, wyszukiwania informacji i systemów wieloprocesorowych.

Dr inż. ANDRZEJ SIEMIŃSKI ukończył w 1977 r. Wydział Informatyki i Zarządzania Politechniki Wrocławskiej. W tym samym roku rozpoczął pracę na tym wydziale jako asystent naukowo-dydaktyczny. W 1984 r. obronił w Instytucie Cybernetyki Technicznej Politechniki Wrocławskiej pracę doktorską nt. metod kompresji danych. Zainteresowania zawodowe obejmują kompresję danych, hypertekstowe systemy wyszukiwania informacji i rozproszone systemy informacyjne. Opublikował ponad 15 prac naukowych.

dobrej jakości drukarki), a nawet może zlecać wykonywanie obliczeń na „obcym" komputerze. Takie dzielenie zasobów było pierwotną motywacją tworzenia sieci. Rozproszenie użytkowników i zasobów w sieci powoduje jednocześnie konieczność komunikowania się komputerowego. Jedną z najprostszych, a jednocześnie najbardziej popularnych usług udostępnianych przez KK jest **poczta elektroniczna.** Pozwala ona przesyłać wiadomości do skrzynek przypisanych użytkownikom. Przyjęty sposób adresowania jest uzależniony od trzech czynników: sieci komputerowej nadawcy, sieci komputerowej odbiorcy i oprogramowania stosowanego do przesyłania wiadomości. Wiadomości są przesyłane nie tylko przez użytkowników, niektóre z nich są generowane przez system. Przykładem wiadomości tego rodzaju jest komunikat o niemożności przesłania wiadomości z powodu np. błędnego adresu. Łatwość podługiwania się oprogramowaniem zarządzającym pocztą i duża szybkość przesyłania wiadomości sprawiają, że usługa ta cieszy się (także w Polsce) bardzo szybko rosnącą popularnością.

Poczta elektroniczna jest najprostszym sposobem KK realizującym połączenia wg schematu: jeden nadawca-jeden odbiorca. Nie różni się jednak w swej istocie w niczym od sposobu działania normalnej poczty. Nową jakość usług można natomiast uzyskać dzięki **listom odbiorców.** Ten rodzaj komunikowania pozwala na przesłanie tej samej wiadomości każdemu z odbiorców wymienionych na liście. Jest to przydatne wówczas, gdy grupa osób pragnie prowadzić dyskusję na określony temat. Skład personalny takiej grupy zmienia się w czasie i dlatego listy te należy uaktualniać, dodają do nich nowych odbiorców i kasując osoby nie zainteresowanie. Niektóre z sieci, np. BITNET, mają oprogramowanie pozwalające na automatyczne zaktualizowanie listy odbiorców. Służy do tego proces o nazwie *LISTSERVER.*

Najbardziej zaawansowanym sposobem komunikowania się komputerowego są konferencje komputerowe. W tym przypadku wielu nadawców może przesyłać widomości wielu odbiorcom. Konferencje komputerowe różnią się od list odbiorców nie tylko liczbą przekazywanych wiadomości, ale i stopniem ich prywatności. W listach wiadomości są przekazywane do konkretnych odbiorców i nowi użytkownicy muszą uzyskać zgodę na wpisanie się na taką listę. Każdy z odbiorców otrzymuje oddzielną wiadomość. W przeciwieństwie do tego, w przypadku konferencji komputerowych wiadomość przeznaczona do rozpowszechniania jest przechowywana przez główny komputer węzła (ang. *host*) i jest ona dostępna dla wszystkich użytkowników do niego przyłączonych. Zgodnie z tą definicją, za konferencje komputerowe będziemy uznawać zarówno systemy wykorzystujące pojedynczy komputer, jak i zestawy wiadomości, dostępne w największych rozproszonych sieciach na świecie.

Z punktu widzenia użytkownika nie ma praktycznie różnic między listami odbiorców a konferencjami komputerowymi. Osoby z nich korzystające będą nazywane w dalszej części artykułu **grupami dyskusyjnymi.** Zachęcamy Czytelników do przyłączenia się do jednej z nich. Wspólne potrzeby zwykły jednoczyć ludzi i dlatego przejawiają oni naturalną tendencję do tworzenia grup o wspólnych zainteresowaniach. Grupy dyskusyjne ułatwiają znacznie ten proces, umożliwiają łatwe porozumiewanie się ich członków niezależnie od miejsca zamieszkania. Wiele z nich jednoczy osoby zainteresowane określoną, szeroką dziedziną, np. superkomputerami lub systemami operacyjnymi, ale nie brak i grup poświęconych bardzo szczegółowym tematom, np. łączących użytkowników programu komunikacyjnego **KERMIT.** W chwili obecnej większość grup dyskusyjnych utworzono dla tematów związanych z informatyką, lecz obserwuje się również stałe zwiększenie udziału grup omawiających zagadnienia społeczne, polityczne lub religijne. W USA istnieje wiele grup poświęconych problemom dydaktycznym (np. jaki wpływ na dydaktykę mają nowe technologie), a nawet elektroniczne czasopismo na temat zdalnego nauczania.

## Wpływ grup dyskusyjnych na proces nauczania

Instytuty badawcze, a także „prawdziwe" konferencje naukowe istnieją ponieważ zachodzi potrzeba zgrupowania w jednym miejscu ludzi rozwiązujących podobne zadania, tak by mogli się oni efektywnie porozumiewać. Komunikowanie się komputerowe sprawia, że znikają ograniczenia wynikające z miejsca zamieszkania lub kosztów przejazdu. Dowolna grupa osób może prowadzić dyskusję na dowolny temat. KK znosi nie tylko te fizyczne ograniczenia. Nowe medium wpływa istotnie na sposób przekazywania informacji. W prawdziwej dyskusji na raz może mówić tylko jedna osoba, KK pozwala zaś na równoczesne przekazywanie wiadomości przez wiele osób. Przyśpiesza to wymianę poglądów, uczestnicy dyskusji nie muszą bowiem czekać na swoją kolej. Ten sposób porozumiewania się umożliwia ponadto skoncentrowanie się na zawartości merytorycznej wypowiedzi. To, co dociera do odbiorców, to tylko jej tekst, a nie forma, nieistotne jest np. jąkanie się nadawcy. Prowadzenie dyskusji w obcym języku jest dla wielu osób łatwiejsze w piśmie niż w mowie. Nie istnieje przy tym groźba przerwania wypowiedzi, ale każdy z wypowiadających się musi sobie zdać sprawę, że zarazem nie ma przymusu wysłuchania wypowiedzi, czytanie jej może być przerwane w dowolnym miejscu, jeśli tylko odbiorca nie jest nią zainteresowany lub też uważa ją za niesłuszną bądź za nieuzasadnioną. Kolejną zaletą, niewidoczną być może na pierwszy rzut oka, jest anonimowość, jaką zapewnia KK. Jest to przydatne wówczas, gdy wymiana poglądów odbywa się między osobami rażąco różniącymi się pozycjami zajmowanymi w hierarchii naukowej, np. między studentami a laureatami nagrody Nobla.

Wszystkie te wymienione powyżej cechy sprawiają, że komunikowanie się komputerowe jest przydatne zwłaszcza dla studentów. Studenci dysponujący komputerem przyłączonym np. do sieci BITNET lub Internet mają dostęp do setek grup dyskusyjnych i mogą brać udział w dyskusjach na praktycznie każdy interesujący ich temat.

Pozwala to na zebranie informacji bardzo trudno dostępnych w inny sposób. Student taki wykorzystuje naturalnie komputer jako narzędzie pracy (np. do przygotowania materiałów do zaliczenia przedmiotu), ale jednocześnie jest on dla niego źródłem wiedzy pozyskiwanej od ekspertów w określonej dziedzinie. Charakterystyczne jest przy tym to, że prawie zawsze wytwarza się silna więź między członkami grupy dyskusyjnej. Członek grupy odkrywa, niejednokrotnie ze zdumieniem, osoby o podobnych zainteresowaniach i problemach, zamieszkałe w innych miastach lub nawet w innych państwach. Osoby często biorące udział w dyskusjach rozpoznają swoje wypowiedzi, nawet jeśli nigdy nie spotykały się osobiście. Komunikowanie się komputerowe jest cennym uzupełnieniem publikacji w czasopismach oraz tradycyjnych konferencji. Można dzięki niemu zarówno znaleźć odpowiedź na szczegółowe pytanie, jak i wyrobić sobie ogólny pogląd na zagadnienie.

W naszym odczuciu, studenci odnoszą wiele korzyści z komunikowania się komputerowego. Nie tylko nabywają wprawy w posługiwaniu się komputerami, ale i zwiększają swoje umiejętności przetwarzania informacji. Zwykle cenią sobie bardzo możność uzyskania porady od eksperta, z którym nie mieliby żadnych szans spotkać się osobiście. Uczestniczą w wielu grupach dyskusyjnych (co zmusza ich do silnej selekcji informacji), często zadają pytania, a po pewnym czasie sami przekazują uwagi.

Porównując studentów korzystających i nie korzystających z tych usług stwierdzamy występowanie między nimi istotnych różnic. Z reguły studenci pierwszej z tych grup:

- mają wyższą średnią ocen;
- posługują się bardziej zaawansowanym słownictwem;
- udzielają bardziej precyzyjnych odpowiedzi na pytania w trakcie egzaminów;
- częściej uczestniczą w dyskusjach podczas zajęć;
- częściej zadają pytania wykraczające poza normalnie przekazywany zakres materiału;
- bez obaw korzystają z usług poczty elektronicznej.

## Grupy dyskusyjne w sieci komputerowej BITNET

W USA istnieje wiele sieci komputerowych wykorzystywanych przez środowiska akademickie. Jedną z nich jest sieć BITNET, o zasięgu ogólnoświatowym. Jej integralną częścią jest EARN, łączący głównie uniwersytety i ośrodki badawcze.

Grupy dyskusyjne w sieci BITNET są zorganizowane głównie na zasadzie list odbiorców. Oznacza to, że aby brać udział w dyskusji, użytkownik musi się najpierw na taką listę wpisać, a gdy nie jest już zainteresowany dalszym uczestniczeniem w dyskusjach – wypisać. Listy te są zakładane i aktualizowane w różnych węzłach sieci BITNET. Niektóre nie mają żadnej, podczas gdy w innych węzłach może być ich wiele. Początkujący użytkownicy sieci mają zwykle wiele kłopotów ze znalezieniem grup dyskusyjnych omawaiających interesujące ich zagadnienia. Aby uzyskać wykaz wszystkich grup dyskusyjnych dostępnych w sieci BITNET, należy wydać polecenie:

*LISTSERV @ any-node*

i przesłać wiadomość:

*LIST global*

Nauczyciel, dysponując pełnym wykazem list odbiorców, może przekazać studentom adresy węzłów, w których są utrzymywane listy dla grup dyskusyjnych omawiających tematy leżące w sferze zainteresowań każdego ze studentów i zgodnie z ich programem studiów.

**Wybrane grupy dyskusyjne dostępne w sieci BITNET**

| Adres | Nazwa |
|---|---|
| ERL-L@TCSVM | EDUCATIONAL RESEARCH LIST (TCSVM) |
| APL-L@UNBVM1 | APL LANGUAGE DISCUSSION |
| APPLE2-L@BROWNVM | APPLE II LIST |
| APPLICAT@DEARN | (PEERED) APPLICATIONS UNDER BITNET |
| BI-L@BINGVMB | BIBLIOGRAPHIC INSTRUCTION DISCUSSION GROUP |
| BITNEWS@BITNIC | (PEERED) BITNET NETWORK NEWS LIST |
| BRS-L@USCVM | BRS/SEARCH FULL TEXT RETRIEVAL SOFTWARE |
| C-L@UIUCVMD | C (LANGUAGE) DISCUSSIONS |
| CARCH-L@IBACSATA | COMPUTER ARCHITECTURE INTERNATIONAL GROUP |
| CHESS-L@GREARN | THE CHESS DISCUSSION LIST |
| CHRISTIA@FINHUTC | PRACTICAL CHRISTIAN LIFE |
| CICS-KER@LEHIIBM1 | CICS-KERMIT DISCUSSION LIST |
| EDUCAI-L@WVNVM | EDUCATIONAL APPLICATIONS OF ART.INTELL. |
| EJCREC@RPIECS | ELECTRONIC JOURNAL OF COMMUNICATION |
| GAMES-L@BROWNVM | (PEERED) COMPUTER GAMES LIST |
| GEOGRAPH@FINHUTC | GEOGRAPHY |
| HISTORY@FINHUTC | HISTORY |
| I-KERMIT@BNANDP11 | (PEERED) INFO-KERMIT DIGEST |
| IR-LIST@IRLEARN | INFORMATION RETRIEVAL DISTRIBUTION LIST |
| JAPAN@FINHUTC | INFO-JAPAN |
| MEDIMAGE@POLYGRAF | MEDICAL IMAGING DISCUSSION LIST |
| MUSIC@FINHUTC | MUSIC-RESEARCH |
| NMBRTHRY@NDSUVM1 | NUMBER THEORY LIST |
| NOVELL@IRLEARN | NOVELL DISCUSSION GROUP |
| ORACLE-L@SBCCVM | ORACLE DATABASE MAILING LIST. |
| PC-EVAL@IRLEARN | PERSONAL COMPUTER EVALUATION |
| PLEARN-L@UBVM | DISCUSSION OF POLISH EARN TOPICS |
| POLAND-L@UBVM | DISCUSSION OF POLISH CULTURE LIST |
| RELIGION@HARVARDA | RELIGIONS DISCUSSION GROUP |
| SCUG-L@MITVMA | SUPERCOMPUTER USERS LIST |
| SF-LIST@JPNKISCI | SF AUTHOR LISTS |
| SILGRA-L@WSUVM1 | SILICON GRAPHICS WORKSTATIONS LIST |
| SIMULA@BITNIC | (PEERED) THE SIMULA LANGUAGE LIST |
| SOFT-ENG@BYUVM | (SOFT-ENG, ARPA DISCUSSIONS) |
| SUPER-L@MCGILL1 | SUPER COMPUTER USERS FORUM |
| TCPIP-L@IRLEARN | TCPIP INFORMATION DISTRIBUTION LIST |
| TR-L@DB0FUB11 | TOKEN RING DISCUSSION-LIST |
| TURBOC-L@TREARN | (PEERED) TURBO C DISCUSSION GROUP. |
| UNIX-SRC@NDSUVM1 | UNIX-SOURCES MAILING LIST |
| VIRUS-L@LEHIIBM1 | VIRUS DISCUSSION LIST |
| VMSLSV-L@UBVM | VAX/VMS LISTSERV DISCUSSION LIST |
| WIN-L@DEARN | X.25 - WISSENSCHAFTSNETZ (TECHN. KOORD.) |

```
pollux.usc.edu(18): mail listserv@unmvm.bitnet
Subject:
SUBSCRIBE CNIDIR-L Kazimierz Kowalski
EOT
pollux.usc.edu(19): logout

From @VM.usc.edu:LISTSERV@UNMVM.BITNET Thu Apr  4 09:15:38 1991
To: kkowalsk@POLLUX.usc.edu

> SUBSCRIBE CNIDIR-L Kazimierz Kowalski
List CNIDIR-L is not open for automatic subscription.
Your request has been forwarded to the list owner(s):
 art stgeorge <STGEORGE@UNMB>
 Art StGeorge <STGEORGE@UNMVM>
 GEORGE BRETT <GHB@UNCECS.EDU>
 george brett <GHB@ECSVAX>
...

From @VM.usc.edu:LISTSERV@UNMVM.BITNET Thu Apr  4 09:16:06 1991
Subject:      Request for subscription to list CNIDIR-L
To: STGEORGE%UNMB.bitnet@bitnet-gw.usc.edu,
      STGEORGE%UNMVM.bitnet@bitnet-gw.usc.edu, GEORGE BRETT

Dear list owner,

  A request for subscription to your distribution list CNIDIR-L
(CNIDIR-L - Coalition for Networked Information Working Group)
has been received on Thursday, April the 4th of
1991 from Kazimierz Kowalski <kkowalsk@POLLUX.USC.EDU>.

  You can, at your discretion, issue the following command to
LISTSERV@UNMVM to add this person to the list: ADD CNIDIR-L
kkowalsk@POLLUX.USC.EDU Kazimierz Kowalski (do not forget the
"PW=list_password" keyword)

Virtually,

  The LISTSERV management

From @VM.usc.edu:LISTSERV@UNMVM.BITNET Thu Apr  4 10:14:20 1991
Subject:      Your subscription to list CNIDIR-L
To: Kazimierz Kowalski <kkowalsk@POLLUX.usc.edu>

Dear networker,

 As of Thursday, April the 4th of 1991, you have been added to
the LISTSERV distribution list CNIDIR-L (CNIDIR-L - Coalition
for Networked Information Working Group) by george brett
<GHB@ECSVAX>.

You may leave the list at any time by sending a "SIGNOFF CNIDIR-L"
command to LISTSERV@UNMVM. Please note that this command must
NOT be sent to the list address (CNIDIR-L@UNMVM) but to the
LISTSERV address (LISTSERV@UNMVM). The amount of
acknowledgement you wish to receive from this list upon
completion of a mailing operation can be changed by means of a
"SET CNIDIR-L option" command, where "option" may be either
"ACK" (mail acknowledgement), "MSGACK" (interactive messages only)
or "NOACK". Contributions sent to this list are automatically
archived. You can obtain a list of the available archive files by
sending an "INDEX CNIDIR-L" command to LISTSERV@UNMVM. These files
can then be retrieved by means of a "GET CNIDIR-L filetype"
command, or using the database search facilities of LISTSERV. Send
an "INFO DATABASE" command for more information on the latter.
Please note that it is presently possible for other people to
determine that you are signed up to the list through the use of
the "REVIEW" command, which returns the network address and name
of all the subscribers. If you do not wish your name to be
available to others in this fashion, just issue a "SET CNIDIR-L
CONCEAL" command. More information on LISTSERV commands can
be found in the "General Introduction guide", which you can
retrieve by sending an "INFO GENINTRO" command to LISTSERV@UNMVM.

Virtually,

  The LISTSERV management
.....

From @VM.usc.edu:CNIDIR-L@UNMVM.BITNET Fri Apr  5 13:06:01 1991
From: Steve Cisler <sac%APPLE.COM@VM.usc.edu>
Subject:      Re: Guess what's happening at CNIDIR-L
To: Kazimierz Kowalski <kkowalsk@POLLUX.usc.edu>

One of the issues we should begin to think about is the
different ways different groups of people attack the problem
of directories. As Glee Cady of Merit pointed out at the last
CNI meeting (some of you were there and some were not, I suppose)
librarians tend to plan, plan, plan and then do. Engineers and
others may tend to plan a bit, do it, change it, perhaps discard
it or go through another iteration. How those two groups
can work together should be interesting. In the second group
I'd like to add commercial firms that are thinking about this problem.
Thinking Machines Inc. has set up a directory of servers that are
serving information using the unix, Max, and Connection Machine code
to the Internet community.
Steve Cisler
Apple Library
10381 Bandley Dr. MS: 8C
Cupertino, CA 95014
sac@apple.com
```

**Wybrane fragmenty wiadomości obejmujące: prośbę o wpisanie na listę, potwierdzenie wpisania i głos w dyskusji**

Po znalezieniu odpowiedniej grupy trzeba wpisać się na listę. W tym celu należy wydać polecenie:

*SEND LISTSERV a adres_węzła*

i przesłać wiadomość:

*SUB nazwa_listy pełny_identyfikator_użytkownika*

przy czym *adres-węzła* określa węzeł, w jakim odpowiednia lista dystrybucyjna jest przechowywana i aktualizowana.

W tabeli umieszczono listę przykładowych grup dyskusyjnych w sieci BITNET wraz z odpowiadającymi im adresami sieciowymi węzłów. Na wydruku zamieszczono fragmenty protokułu zawierające: przesłanie wiadomości z prośbą o wpisanie na listę, otrzymanie potwierdzenia wpisania oraz uzyskane fragmenty dyskusji.

* * *

Rola odgrywana przez komputery w procesie dydaktycznym nie jest jeszcze dokładnie określona. Wydaje się jednak pewne, że w najbliższej przyszłości komputery zmienią sposób nauczania, wpływając w ten sposób pośrednio na całe społeczeństwo. Dlatego też należy przeprowadzić liczne badania mające na celu określenie korzyści, jakie wynikają dla studentów z komunikowania się komputerowego. Duża liczba grup dyskusyjnych i obszerna tematyka przez nie pokrywana umożliwia ich wykorzystanie praktycznie na każdej specjalności.

Na obecnym etapie rozwoju komunikowania się komputerowego, nawet w najbardziej rozwiniętych krajach, przedwczesne jest jeszcze naturalnie traktowanie sieci komputerowych jako podstawowego źródła informacji zastępującego np. książki lub czasopisma. Niemniej, już obecnie stanowią one cenne uzupełnienie tradycyjnej naukowej wymiany informacji.

LITERATURA


[1] Featheringham T.: Teleconferences: the Message is a Meeting. Data Communications, No. 6(7), 1977, pp. 37–41

[2] Hiltz S. R.: Communication Medium. Technological Forecasting and Social Change, No. 10(3), 1977, pp. 255–238

[3] Hiltz S. R., Turoff M.: The Evaluation of User Behavior in a Computerized Conferencing System. Communictations of the ACM, No. 24(11), 1981, pp. 739–751

[4] Hiltz S. R., Turoff M.: Structuring Computer-Mediated Communication Systems to Avoid Information Overload. Communications of the ACM, No. 28(7), 1985, pp. 680–689

[5] Kennett K. F.: Computers in Education: Rear Vision Thinking of Golden Opportunities for the Future. Education No. 110(3), 1990, pp. 403–409

[6] LaQuey T.: Networks for Academics. Academic Computing, November, 1989, pp. 32–65

[7] Palme J.: The COM and PortaCom Computer Conference Systems. Stockholm University Computing Centre-QZ, Sweden, 1 August, 1984

[8] Price C. R.: Conferencing via Computer: Cost Effective Communication for the Era of Forced Choice. In: Turoff M. (Ed.): The Delphi Method. Addison-Wesley, Reading (MA), 1975

[9] Quarteman J. S.: The Matrix. A Digital Press, 1990

[10] Ricart G.: Slowing the Big Bang of Computer Networking. Academic Computing, November, 1989, pp. 28-55

[11] Richards B. F.: Should Instructional Designers Design Expert System. Educational Research and Development, No. 37(3), 1989, pp. 63–71

[12] Sirc G. M.: Learning to wirte on LAN. The Journal, No. 15(8), 1989, pp. 99–104

[13] Tanenbaum A. S.: Computer Networks. Prentice-Hall. Englewood Cliffs (NJ), 1988

[14] Uhlis G. E.: On William J., Bennett U. S. Secretary of Education. Education, No. 108, 1987, pp. 152–154

[15] Ullmer G. E.: Hign-Tech Instructional Development: It's the Thought that Counts. Educational Technology Research and Development, No. 37(3), 1987, pp.95–101

[16] Video Term Papers Take the Place of Traditional Written Reports. The Journal, No. 15(8), 1988, pp. 68–74

[17] Zuboff S.: In the Age of Smart Machine: the Future of Word and Power. BASIC Books, New York, (NY), 1988.


## Wzrost ceny INFORMATYKI

Informujemy naszych Czytelników, że od bieżącego numeru cena INFORMATYKI wzrasta o 20% – do kwoty 18 000 zł za jeden egzemplarz. Podwyżki tej dokonujemy dopiero po 15 miesiących od ustalenia poprzedniej ceny. Obecny wzrost ceny jest procentowo prawie pięciokrotnie mniejszy od oficjalnej stopy inflacji we wspomnianym okresie, co oznacza, że INFORMATYKA relatywnie staniała. Jesteśmy przekonani, że w tym świetle podwyżka spotka się z pełnym zrozumieniem naszych odbiorców.

Nasi dotychczasowi prenumeratorzy już otrzymali lub wkrótce otrzymają powiadomienia o konieczności uzupełnienia należności (27 000 zł w przypadku prenumeraty rocznej). W dalszym ciągu utrzymujemy prenumeratę ulgową (13 500 zł za egzemplarz) dla uczącej się młodzieży oraz członków stowarzyszeń inżynierskich zrzeszonych w Naczelnej Organizacji Technicznej.

Cena INFORMATYKI jest w decydującym stopniu uzależniona od wielkości nakładu. Apelujemy więc gorąco do naszych dotychczasowych Czytelników o propagowanie prenumeraty naszego miesięcznika w całym środowisku profesjonalnym, ponieważ stwierdziliśmy, że wskutek zaistniałych zmian strukturalnych oraz olbrzymiego rozrostu tego środowiska INFORMATYKA mimo swej nieprzerwanej 27-letniej tradycji jest w niektórych kręgach tego środowiska praktycznie nieznana. Przyczyniło się do tego całkowite zniknięcie w ubiegłym roku czasopisma ze sprzedaży kioskowej, wskutek kryzysu, jaki przeżywa dotychczasowy monopospolita RUCH. Dlatego obecnie jedyną drogą regularnego otrzymywania INFORMATYKI jest jej zaprenumerowanie lub kupienie w Redakcji oraz dziale handlowym SIGMY. Zapytania prosimy kierować bezpośrednio do **Redakcji, Pl. Inwalidów 10, p. 104, 01–552 Warszawa, tel. 39-14-34.**


meditronik SPÓŁKA z o.o.

UMC UNITED MICROELECTRONICS CORPORATION
- UKŁADY PAMIĘCI
- UKŁADY KOMPUTEROWE
- UKŁADY KOMUNIKACYJNE I KOMERCYJNE

PRZEDSTAWICIELSTWO

HEWLETT PACKARD COMPONENTS
- TRANSOPTORY
- WSKAŹNIKI ŚWIETLNE
- WYŚWIETLACZE LED
- PRODUKTY KODÓW KRESKOWYCH
- KONTROLERY I CZUJNIKI RUCHU
- TECHNIKA ŚWIATŁOWODOWA
- ELEMENTY W.CZ. I MIKROFALOWE
- PODZESPOŁY DO MONTAŻU POWIERZCHNIOWEGO (SMD)

DYSTRYBUCJA

BOURNS
- POTENCJOMETRY TRIMPOT
- HYBRYDY REZYSTOROWE
- REZYSTORY SUBMINIATUROWE
- BEZPIECZNIKI MULTIFUSE
- POTENCJOMETRY PRECYZYJNE
- POTENCJOMETRY PANELI CZOŁOWYCH I KODERY
- CEWKI I TRANSFORMATORY
- CZUJNIKI CIŚNIENIA, POŁOŻENIA I PRZYŚPIESZENIA

PMI

DYSTRYBUCJA

- CZĘŚCI ELEKTRONICZNE
- KOMPUTERY PS/1,PS/2
- DRUKARKI HP
- INSTALACJE SIECI KOMPUTEROWYCH

Partnerzy handlowi: ANALOG DEVICES, ITT, MOTOROLA, SAMSUNG, TELEFUNKEN i inni

IBM Business Partner

meditronik sp. z o.o.

00-194 Warszawa, ul. Dzika 4
tel. (02) 6352263, 6352264
fax (02) 6352195, tlx 816075

SO/575/91

WIESŁAW PRZETACZNIK
Raciborska Fabryka Kotłów RAFAKO
Racibórz

# Graficzne stacje robocze SPARCstation

Wejście na rynek a następnie opanowanie go przez procesory o architekturze RISC stało się faktem. Pomimo marudzenia niektórych malkontentów (producentów standardowych CISC), procesory te nie tylko znalazły chętnych użytkowników, ale przede wszystkim spełniły ich oczekiwania. Okazało się bowiem, że wcześniejsze obietnice i zapowiedzi mogą być zrealizowane. Koncepcja okrojonej listy instrukcji (zmniejszony czas przeszukiwania drzewa decyzyjnego instrukcji), wykonywanych w pojedynczym cyklu zegarowym, zastosowana do architektury procesora radykalnie zwiększyła sprawność komputera. Dzieje się to kosztem wydłużenia kodu źródłowego programu, ale w efekcie jest opłacalne.

SUN Microsystems, firma wyłoniona z z renomowanego Standford University (*Standford University Network*) i największy dziś w świecie producent stacji roboczych (30%) opracował własną wersję procesora typu RISC o handlowej nazwie SPARC (*Scalable Procesor Architecture*). Pierwsza stacja robocza z tym procesorem – SUN-4, została już zastąpiona przez nowe konstrukcje o wspólnej nazwie SPARCstation. Rodzinę tę tworzy kilka modeli o zróżnicowanych parametrach. W niniejszym artykule omówione zostaną tylko graficzne stacje robocze, pominięte natomiast – modele przeznaczone do zarządzania zasobami w sieci.

## SPARCstation SLC

Jest to najmniejszy bezdyskowy model rodziny, projektowany jako desktop, z całą jednostkę centralną mieszczącą się w obudowie monitora. Uzupełnienie stacji stanowi 107-klawiszowa, niskoprofilowa klawiatura i trójklawiszowa mysz optyczna.

Chłodzony konwencjonalny monitor zawiera pięć wbudowanych portów:

- łącze sieciowe Ethernet (15-pinowe typu *D*), zapewnia przy zastosowaniu kabla koncentrycznego szybkość transmisji 10 Mb/sekundę,
- dwa asynchroniczne porty szeregowe RS-232C (25-pinowe typu *D*) do przyłączenia wolnych urządzeń peryferyjnych; pierwszy – do dwukierunkowej transmisji, np. modem, drugi – tylko do wyprowadzania danych, np. drukarka,
- adapter magistrali SCSI (ang. *Small Computer System Interface*), (łącze typu SCSI-2) do przyłączenia pamięci masowych: dysk twardy, pamięć taśmowa, CD-ROM;
- port wejścia-wyjścia audio, 8-bitowy, uzupełnieniem jest wewnętrzny głośnik.

Ukierunkowanie na zastosowanie graficzne zaowocowało monitorem o następujących właściwściach:

- typ monochromatyczny,
- rozdzielczość 1152 × 900 punktów,
- zagęszczenie 100 punktów na cel,
- współczynnik kształtu obrazu: 1:1 (ang. *pixel aspect ratio*),
- częstotliwość odświeżania 66 MHz, odświeżanie bez przeplotu,
- szerokość pasma: 93 MHz,
- zasilanie: 110–120/200–240 V automatyczne przełączalne,
- sterowanie jasnością, kontrastem: odchylanie poziome i pionowe, środkowanie poziome i pionowe.

Monitor jest umieszczony na ruchomej i obrotowej podstawie. Duża rozdzielczość pozwala realizować zaawansowane prace graficzne i efektywne wyświetlanie jednocześnie kilku okien.

Wewnątrz monitora umieszczono pionowo płytę główną komputera. Jej podstawową część zajmuje układ procesora centralnego, który tworzą:

- procesor całkowitoliczbowy SPARC RISC (ang. *SPARC integer unit*);
- procesor zmiennoprzecinkowy SPARC IEEE standard 754 (ang. *SPARC floating-point unit*);
- pamięć notatnikowa SPARC cache o pojemności 64 KB.

Przy częstotliwości taktowania 20 MHz moc obliczeniowa wg danych producenta osiąga:

- dla operacji całkowitoliczbowych: 12,5 Dhrystone MIPS,
- dla operacji zmiennoprzecinkowych: 1,2 MFLOPS DP LINPACK.

Pamięć operacyjna ma standardowo pojemność 8 MB z możliwością rozszerzenia do 16 MB. Zastosowano układy typu 4 MB SIMM. Systemem pamięci zarządza moduł MMU (*SPARC Memory Management Unit*), zastosowany i sprawdzony już w rodzinie SUN-4. Pobiera on z procesora centralnego adres wirtualny i przekształca go na fizyczny adres pamięci operacyjnej, układów wejścia-wyjścia, pamięci obrazu, układów obsługi magistrali. Standardowo wszystkie te zespoły są traktowane jednakowo.

Główną arterię przesyłową całej rodziny SPARCstation stanowi magistrala SBus, która zawiera 32 linie danych i 32 linie adresowe. Skąpość miejsca wewnątrz monitora nie zezwala na rozbudowę modelu – brak jest łączy krawędziowych (ang. *slots*) na karty rozszerzające.

**SPARCstation SLC** jest samodzielną stacją roboczą bez wewnętrznej pamięci masowej. Szybki interfejs SCSI pozwala bezpośrednio przyłączyć następujące moduły konfiguracji typu desktop:

- dysk twardy standardu 3,5″ o pojemności 104 MB; średni czas przeszukiwania (ang. *average seek time*) – 22 ms; szybkość

---

Mgr inż. **WIESŁAW PRZETACZNIK** ukończył w 1983 r. wydział Automatyki, Elektroniki i Informatyki (kierunek Informatyka) na Politechnice Śląskiej w Gliwicach o specjalności budowy i oprogramowania maszyn matematycznych. Od 1986 r. pracuje w Biurze Projektowym Urządzeń Energetycznych „RAFAKO". Zajmuje się komputerowym wspomaganiem prac inżynieryjnych CAE (obliczenia numeryczne, wspomaganie prac projektowych CAD).

transmisji (ang. *data rate*) – 1,5 MB/s;
- dysk optyczny typu ROM (tylko odczyt) o średnicy 5,25″ i pojemności 644 MB; średni czas przeszukiwania 400 ms; szybkość transmisji 1,8 MB/s.

Możliwe jest również sterowanie następującymi modułami zewnętrznymi:
- dyskiem twardym standardu 5,25″ o pojemności 327 MB; średni czas przeszukiwania 16,5 ms; szybkość transmisji 1,5 MB/s;
- dyskiem twardym standardu 5,25″ o pojemności 669 MB; średni czas przeszukiwania 16 ms; szybkość transmisji 1,8 MB/s;
- pamięcią taśmową standardu 1/4″ typu streamer – QIC-150, o pojemności 150 MB.

Pakiety programowe na stacje robocze wyszły już poza typowe zastosowania w konstrukcyjnych pracach inżynieryjnych. Producent przewiduje zastosowanie **SPARCstation SLC** w dziedzinach:
- inżynieria oprogramowania (ang *software enginnering*),
- mała poligrafia (ang. *electronic publishing*),
- zarządzanie bazami (ang. *database management*),
- systemy wspomagania podejmowania decyzji (ang. *decision support*),
- automatyzacja prac biurowych (ang. *office automation*).

Przy omawianiu kolejnych modeli SPARCstation zostaną opisane tylko istotne rozszerzenia w stosunku do modelu podstawowego.

## SPARCstation IPC i SPARCstation IPC GX

Układ procesora centralnego tworzą:
– procesor całkowitoliczbowy SPARC RISC, taktowany zegarem 25 MHz,
– procesor zmiennoprzecinkowy SPARC IEEE standard 764,
– pamięć notatnikowa SPARC cache o pojemności 64 KB.

Zwiększona częstotliwość zegara taktującego pozwoliła osiągnąć:
- dla operacji całkowitoliczbowych 15,8 Dhrystone MIPS,
- dla operacji zmiennoprzecinkowych 1,7 MFLOPS DP LINPACK.

Standardową wielkość pamięci operacyjnej – 8 MB (przy zastosowaniu układów 1 MB SIMM), można zwiększyć do 12 MB (układy 1 MB), 24 MB (układy 4 MB), lub 48 MB (12 układów 4 MB SIMM).

Konstrukcja stacji odzwierciedla filozofię otwartej architektury. Stąd rozbudowę konfiguracji można realizować przez łatwe montowanie dodatkowych kart (również innych producentów) w dwóch złączach umożliwiających dostęp do magistrali SBUS.

Model IPC został już wyposażony w następującą wewnętrzną pamięć masową (ang. *internal storage*):
- stację dysków elastycznych standardu 3,5″ o pojemności 1,44 MB/720 KB; format zapisu mikrodyskietek zgodny jest z MS DOS;
- dysk twardy standardu 3,5″ o pojemności po sformatowaniu 207 MB; średni czas przeszukiwania 16 ms.

Stacja IPC może być rozbudowana o następujące rodzaje masowej pamięci podręcznej (w obudowie typu destop):
- dysk typu Winchester standardu 3,5″ o pojemności po sformatowaniu 207 MB; średni czas przeszukiwania 16 ms, szybkość transmisji pakietu 4 MB;
- pamięć taśmowa standardu 1/4″ typu streamer QIC-150 o pojemności kasety 150 MB;
- dysk optyczny standardu 5,25″ o pojemności 644 MB.

Szybkość transmisji danych 1,2 MB/s.

Wbudowany sterownik pozwala zarządzać następującymi modułami pamięci zewnętrznej (ang. *external storage modules*):
- dyskiem typu Winchester standardu 5,25″ o pojemności po sformatowaniu 669 MB, średni czas przeszukiwania 16 ms, szybkość transmisji pakietu danych 4 MB/s,
- pamięć taśmowa standardu 1/4″ typu streamer QIC-150; pojemność kasety po sformatowaniu 150 MB,
- pamięć taśmowa typu „helical scan" standardu 8 mm o gigantycznej pojemności GB.

W zależności od potrzeb użytkownika model **SPARCstation IPC** może być wyposażony w jeden z trzech typów monitorów o dużym kontraście. Każdy z nich ma rozdzielczość 1152 × 900 piksli oraz współczynnik kształtu obrazu 1:1;
- kolorowy o przekątnej ekranu 16″; zagęszczenie 100 punktów/cal, częstotliwość odświeżania 76 lub 66 Hz, bez przeplotu, szerokość pasma 93 MHz, liczba barw 256 z palety ponad 16,7 mln;
- monochromatyczny o przekątnej 19″; zagęszczenie 84 punktów/cal, częstotliwość odświeżania 66 Hz, bez przelotu, szerokość pasma 93 MHz;
- monochromatyczny o przekątnej 17″; zagęszczenie 100 punktów/cal; częstotliwość odświeżania 76 Hz, bez przeplotu, szerokość pasma 93 MHz.

**Model SPARCstation IPC GX** oferuje użytkownikowi wbudowany i całkowicie zintegrowany akcelerator graficzny, który w znaczący sposób zwiększa możliwości stacji. Predysponuje to do stosowania tego modelu w szeroko rozumianych systemach komputerowego wspomagania prac inżynieryjnych CAE (*Computer Aided Engineering*), a w szczególności w systemach wspomagania konstruowania CAD (*Computer Aided Design*) oraz małej poligrafii. Sprzętowa realizacja operacji graficznych w znaczący sposób przyspiesza ich wyświetlanie na ekranie monitora. Daje to możliwość swobodnego operowania grafiką dwu- i trówymiarową, zarówno siatkową (ang. *wireframe graphics*), modelowaniem powierzchniowym (ang. *surface modelling*) oraz bryłowym (ang. *solid modelling*). Sprzętowo są wykonywane takie operacje, jak: wygładzanie linii (ang. *antialiasing*) i zmniejszenie jasności w perspektywie (ang. *depth-cueing*), usuwanie niewidocznych linii (ang. *z-buffering*), obcinanie rysunków (ang. *clipping*). Obliczenia graficzne są fizycznie oddzielone od normalnych obliczeń numerycznych realizowanych w procesorze centralnym, co w ogromnym stopniu zwiększa efektywność. Akcelerator graficzny nie tylko wspomaga podstawowe operacje graficzne, ale również umożliwia płynne przekształcanie całych obrazów, a więc ich animację, uwzględniając np. zmiany oświetlenia, jasności, cienie itp.

**Model SPARCstation IPC GX** jest oferowany z następującymi monitorami o dużej rozdzielczości (1152 × 900 piksli) i współczynnikiem kształtu obrazu 1:1:
- monitorem kolorowym o przekątnej ekranu 16″: zagęszczenie 100 punktów/cal, częstotliwość odświeżania 76 lub 66 Hz, bez przeplotu;
- monitorem kolorowym o przekątnej 19″: zagęszczenie 84 punktów/cal, częstotliwość odświeżania 66 Hz, bez przeplotu;
- monitorem monochromatycznym o przekątnej 19″: zagęszczenie 84 punktów/cal, częstotliwość odświeżania 76 Hz, bez przeplotu.

Duża rozdzielczość i obszerny zakres kolorów na ekranie zapewnia wyjątkowy realizm nawet złożonych modeli.

## SPARCstation 1+ i SPARCstation 1 + GX

Model **SPARCstation 1 +** został zaprojektowany z przeznaczeniem do systemów: wspomagania inżynierii oprogramowania CASE (*Computer Aided Software Engineering*), wspomagania zintegrowanego wytwarzania CIM (*Computer Integrated Manufacturing*), sztucznej inteligencji AI (*Artificial Intelligence*), finansowych i małej poligrafii.

Procesor taktowany zegarem 25 MHz daje sprawność taką jak w modelu IPC. Standardowa pamięć 8 MB może być zwiększona przy zastosowaniu układów 4 MB SIMM nawet pięciokrotnie.

W obudowie jednostki centralnej mieszczą się napędy pamięci masowej:
- dwa dyski twarde typu Winchester standardu 3,5″ o pojemności po sformatowaniu 104 MB każdy, średni czas przeszukiwania 22 ms, szybkość transmisji 1,2 MB/s;
- opcjonalnie stacja dysków elastycznych standardu 3,5″ o pojemności 1,44 MB.

Pamięć tę można rozbudować o **moduły wewnętrzne** typu desktop:
- dodatkowy dysk twardy o pojemności 104 MB,
- pamięć taśmową QIC-150,
- dysk optyczny o pojemności 644 MB,

oraz **moduły zewnętrzne,**
- do dwóch dysków twardych standardu 5,25″ o pojemności 327 MB każdy, średnim czasie przeszukiwania 16,5 ms i szybkości transmisji 1,5 MB/s,
- dysk twardy standardu 5,25″ o pojemności 669 MB, średnim czasie przeszukiwania 16 ms i szybkości transmisji 1,8 MB/s;
- pamięć taśmową QIC-150,
- pamięć taśmową typu „helican scan" o pojemności 2,3 GB.

Do stacji są oferowane zarówno monitory monochromatyczne o przekątnej ekranu 17″ lub 19″, jak i kolorowe o przekątnej ekranu 16″ lub 19″.

Wszyskie monitory dysponują standardową dla SPARCstation rozdzielczością 1152 × 900 punktów. Zagęszczenie wynosi 100 (16″ i 17″) oraz 84 (19″) punktów/cal. Odświeżanie realizowane jest bez przeplotu, z częstotliwością 66 MHz. Monitory kolorowe pozwalają wyświetlać 256 barw z palety ponad 16,7 mln.

Magistrala SBUS jest wyposażona w trzy złącza umożliwiające łatwą rozbudowę konfiguracji, np. o karty wejścia-wyjścia, wysokowydajną grafikę, karty sieciowe itp.

Akcelerator w modelu **SPARCstation 1 + GX** przyspiesza znacząco operacje graficzne i zapewnia zobrazowanie na ekranie:
- 540 000 wektorów dwuwymiarowych/s,
- 270 000 wektorów trójwymiarowych/s,
- czyszczenie ekranu – 123/s,
- w trybie tekstowym – do 81 000 znaków/s.

## SPARCstation 2

Jest to najbardziej wydajna stacja rodziny i podobnie jak i pozostałe modele ma obudowę typu desktop. Ogromna szybkość przetwarzania, szybkie porty wejścia-wyjścia, duże możliwości rozszerzeń i szeroka gama konfiguracji graficznych czynią ją szczególnie atrakcyjną dla użytkowników. Oprócz podstawowego modelu **SPARCstation 2GX** (z akceleratorem graficznym), oferowane są modele **SPARCstation 2GS** oraz **SPARCstation 2GT.** Według opinii producenta **SPARCstation 2GS** spełnia oczekiwania najbardziej wymagającego użytkownika interaktywnych systemów modelowania bryłowego (trójwymiarowego), zaś **SPARCstation 2GT** jeszcze przewyższa swego poprzednika umożliwiając realizację najwyższej jakości grafiki 3D i wysokiej jakości animacji komputerowej (ang. *computer animation*).

Procesor centralny, taktowany zegarem 40 MHz, osiąga w testach następujące szybkości:
- dla operacji całkowitoliczbowych – 28,5 Dhrystone MIPS,
- dla operacji zmiennoprzecinkowych – 4,2 MFLOPS DP LINPACK,

Szybkie przetwarzanie wspomaga 64 KB pamięci notatnikowej.

Standardowa pamięć półprzewodnikowa o pojemności 16 MB może być rozbudowana na płycie głównej do 64 MB lub nawet 96 MB przy zastosowaniu 32-megabajtowej karty dodatkowej. W tym celu wykorzystuje się jedno z trzech złączy krawędziowych magistrali SBus. Rozszerzony algorytm zarządzania pamięcią spełnia wymagania pracy z wieloma procesami i dużą pamięcią wirtualną.

Wewnętrzną pamięć masową stanowią: stacja dysków elastycznych standardu 3,5″ (podobnie jak w poprzednich modelach zgodna z formatem MS DOS – 1,44 MB) oraz dysk twardy o pojemności 207 MB (opcjonalnie dwa takie dyski). Pamięć tę można rozszerzyć o moduły typu desktop oraz moduły zewnętrzne o łącznej pojemności do 7,6 GB i szybkości i transmisji 5 MB/s.

Własności graficzne poszczególnych modeli są zróżnicowane. Model **SPARCstation 2GX,** wyposażony w akcelerator graficzny, pozwala uzyskać:
- 550 000 wektorów dwuwymiarowych/s,
- 240 000 wektorów trójwymiarowych/s.

W znaczący sposób są przyspieszone również operacje wyświetlania i przewijania tekstu oraz operowania okienkami. 8-bitowa grafika pozwala operować jednocześnie 256 barwami spośród dostępnych 16 777 216.

Model **SPARCstation 2GS** umożliwia wykreślenie w ciągu sekundy 150 000 trówymiarowy wektorów lub do 20 000 cieniowanych wieloboków (ang. *shaded polygons*), przy ośmiu źródłach światła jednocześnie. Rozszerzenie pamięci ekranu do 24 bitów/piksel umożliwia wyświetlenie jednocześnie 16,7 mln barw, co zapewnia bardzo realistyczny obraz. 12-bitowe podwójne buforowanie zwiększa płynność animacji, natomiast 16-bitowe buforwanie linii odciętych umowżliwia dynamiczne usuwanie powierzchni niewidocznych.

Najwyższą jakościowo grafikę trójwymiarową oferuje model **SPARCstation 2GT,** mianowicie:
- 500 000 trójwymiarowych wektorów/s,
- 300 000 „wygładzonych" trójwymiarowych wektorów/s (ang. *anti-aliased 3-D vectors*),
- 100 000 cieniowanych wieloboków/s.

Właściwość „wygładzania" linii pozwala korygować wady zobrazowania spowodowane podziałem rastra. Operacja taka jest realizowana sprzętowo w stosunku do wektorów, wieloboków oraz tekstu.

Porównanie modeli roboczych stacji graficznych SPARCstation

| Rodzina | SPARCstation | | | |
|---|---|---|---|---|
| Model | SLC | IPC/IPC GX | 1 + /1 + GX | 2/2 GX/2 GS/2 GT |
| Procesor: | | | | |
| – całkowitoliczbowy | SPARC RISC | SPARC RISC | SPARC RISC | SPARC RISC |
| – zmiennoprzecinkowy | SPARC IEEE 754 | SPARC IEEE 754 | SPARC IEEE 754 | SPARC IEEE 754 |
| Częstotliwość zegara | 20 MHz | 25 MHz | 25 MHz | 40 MHz |
| Wydajność | | | | |
| – operacji całkowitoliczbowych | 12,5 MIPS | 15,8 MIPS | 15,8 MIPS | 28,5 MIPS |
| – operacji zmiennoprzecinkowych | 1,2 MFLOPS | 1,7 MFLOPS | 1,7 MFLOPS | 4,2 MFLOPS |
| Pamięć buforowa | 64 KB | 64 KB | 64 KB | 64 KB |
| Pamięć operacyjna: | | | | |
| – standardowa | 8 MB | 8 MB | 8 MB | 16 MB |
| – maksymalna | 16 MB | 24 MB | 40 MB | 64 (96) MB |
| Magistrala | SBus | SBus | SBus | SBus |
| – l.adresowe/l.danych | 32/32 | 32/32 | 32/32 | 32/32 |
| Liczba slotów | 0 | 2 | 3 | 3 |
| Porty wejściowe-wyjściowe | Ethernet<br>2 × RS323-C<br>SCSI-2<br>Audio I/O | Ethernet<br>2 × RS323-C<br>SCSI-2<br>Audio I/O | Ethernet<br>2 × RS323-C<br>SCSI-2<br>Audio I/O | Ethernet<br>2 × RS323-C<br>SCSI-2<br>Audio I/O |
| Pamięć masowa (typ format pojemność): | | | | |
| – wewnętrzna | – | FDD/3,5″/1,44 MB<br>HD/3,5″/207 MB | FDD/3,5″/1,44 MB<br>HD/3,5″/2 × 104 MB | FDD/3,5″/1,44 MB<br>HD/3,5″/207 MB |
| – desktop | HD/3,5″/104 MB<br>CD/5,25″/644 MB<br>TD/1/4″/150 MB | HD/3,5″/207 MB<br>CD/5,25″/644 MB<br>TD/1/4″/150 MB | HD/3,5″/104 MB<br>CD/5,25″/644/MB<br>TD/1/4″/150 MB | HD/3,5″/207 MB<br>CD/5,25″/644 MB<br>TD/1/4″/150 MB |
| – zewnętrzna | HD/5,25″/327 MB<br>HD/5,25″/669 MB<br>TD/1/4″/150 MB<br>TD/8 mm/2,3 GB | HD/5,25″/669 MB<br>TD/1/4″/150 MB<br>TD/8 mm/2,3 GB | HD/5,25″/327 MB<br>HD/5,25″/669 MB<br>TD/1/4″/150 MB<br>TD/8 mm/2,3 GB | HD/5,25″/327 MB<br>HD/5,25″/669 MB<br>TD/1/4″/150 MB<br>TD/8 mm/2,3 GB |
| Monitor | | | | |
| – typ | 17″ mono | 17″, 19″ mono<br>16″, 19″ kolor | 17″, 19″ mono<br>16″, 19″ kolor | 19″ mono<br>16″, 19″, 21″ kolor |
| – rozdzielczość | 1152 × 900 | 1152 × 900 | 1152 × 900 | 1152 × 900, 1280 × 1024 |
| – współczynnik kształtu obrazu | 1:1 | 1:1 | 1:1 | 1:1 |
| – odświeżanie | bez przeplotu | bez przeplotu | bez przeplotu | bez przeplotu |

**Symbole pamięci masowych:**
**FDD – dysk elastyczny (ang.** *floppy disk drive*),
**HD – dysk twardy (ang.** *hard disk*),
**CD – pamięć optyczna CD ROM (ang.** *compact disk read only memory*),
**TD – pamięć taśmowa (ang.** *tape drive*)

Cieniowanie brył przestrzennych można uzyskać przy jednoczesnym oświetleniu ich z 32 źródeł światła.

Płynność animacji i szybkość ukrywania niewidocznych powierzchni wzrosły dzięki zwiększeniu buforowania do 24 bitów. Zmniejszenie jasności w perspektywie poprawiło w znaczący sposób wizualne tworzenie perspektywy.

Oprócz standardowych monitorów: monochromatycznego 19″ oraz kolorowych 16″ i 19″ (podobnie jak we wcześniejszych modelach), **SPARCstation 2GT** można dodatkowo wyposażyć w kolorowy monitor o przekątnej 21″ i następujących parametrach:

- rozdzielczość: 1280 × 1024 piksli,
- współczynnik kształtu obrazu: 1:1,
- zagęszczenie: 87 punktów/cal,
- częstotliwość odświeżania: 76 MHz, bez przeplotu.

Wszystkie modele **SPARCstation 2** mogą być wyposażone w takie same moduły pamięci masowej jak pozostałe modele. Łączna pojemność możliwych do przyłączenia zewnętrznych dysków dochodzi do 7,6 GB. Szybki czas dostępu (5 MB/s) zapewniają wówczas zawarty na płycie głównej oraz dwa dodatkowe synchroniczne sterowniki SCSI.

Zestawienie porównawcze wszystkich modeli roboczych stacji graficznych SPARCstation zawiera tabela.

LITERATURA


[1] Rybnik J., Solak J.: Stacje robocze Sun-3. Informatyka nr 3, 1989
[2] SPARCstation SLC. SUN Microsystems, Inc. 1990
[3] SPARCstation IPC. SUN Microsystems, Inc. 1990
[4] SPARCstation IPC. and SPARCstation IPC GX. SUN Microsystems, Inc. 1990
[5] SPARCstation 1 + and SPARCstation 1 + GX. SUN Microsystems, Inc. 1990
[6] SPARCstation 2 Series. SUN Microsystems. Inc. 1990
[7] SunLink: Sun'a Family of Data Communication Products. SUN Microsystems. Inc. 1986.



AITECH ARTIFICIAL INTELLIGENCE LABORATORY

Katowice 14, skr. poczt. 2804, tel.: 150-27-91

**PC-SHELL 1.0**

NARZĘDZIE DO BUDOWY SYSTEMÓW EKSPERTOWYCH

Rodzaje zastosowań:
* systemy doradcze * systemy wspomagania decyzji
* dydaktyka * szybka budowa prototypów

Klasy problemów:
* diagnostyka * klasyfikacja * interpretacja danych

Implementacja: IBM XT, AT i zgodne, DOS

Cena kompletu (dyskietka + dokumentacja):
* 480 000 zł – wersja pełna (do 5000 reguł, przeciętnie 1500–2500),
* 240 000 zł – wersja „Small" (do 200 reguł).

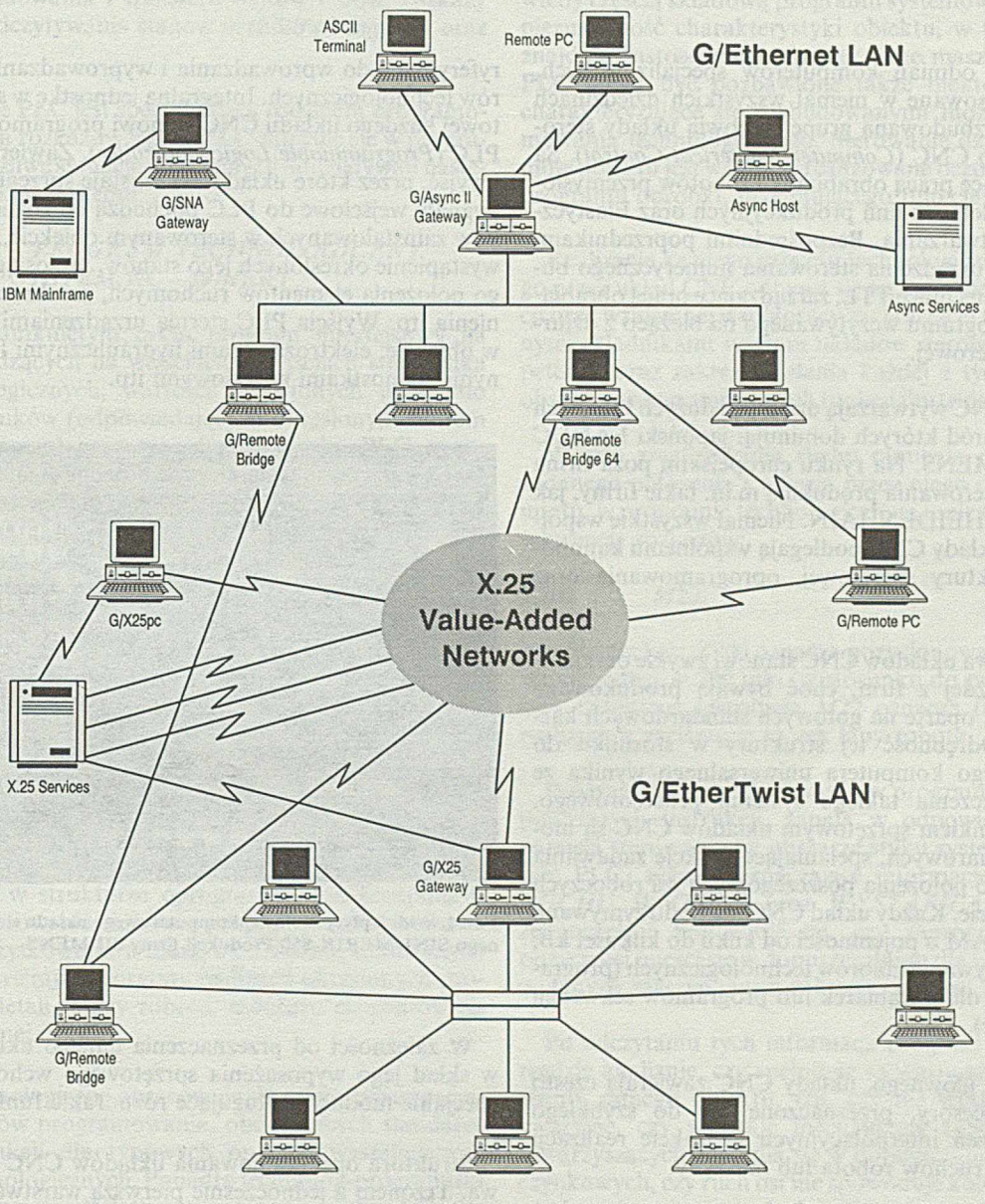
Zniżki przy zakupie wyższych wersji systemu!

Nr konta: 927134-134633-136-61,
Bank Spółdzielczy w Katowicach

246/92

# Produkty sieciowe LAN i WAN

Gateway communications, inc.




BEZKONKURENCYJNE PRODUKTY
SIECI KOMPUTEROWYCH NAGRADZANE PRZEZ:

**PC MAGAZINE, LAN MAGAZINE, INFO WORLD**

OFERUJE AUTORYZOWANY DYSTRYBUTOR:

MIKROB

**P.W.P.T. MIKROB SP. Z O.O.**

**20-346 Lublin, ul. Długa 5**

**Tel. (0-81) 420-61, faks 415-43, teleks 643776 MIKRO PL**

*0/15/91*

Na życzenie wysyłamy katalog produktów. Atrakcyjny program współpracy dla dealer'ów.

BOGDAN KIETZMAN
Fabryka Obrabiarek MECHANICY
Pruszków

# Komputerowe układy sterowania numerycznego

Spośród licznych odmian komputerów specjalizowanych, jakie obecnie są stosowane w niemal wszystkich dziedzinach życia, szczególnie rozbudowaną grupę stanowią układy sterowania numerycznego CNC (*Computer Numerical Control*). Są to urządzenia sterujące pracą obrabiarek i robotów przemysłowych, a także kompletnych linii produkcyjnych oraz Elastycznych Systemów Wytwarzania. Bezpośrednimi poprzednikami układów CNC były urządzenia sterowania numerycznego budowane zazwyczaj w technice TTL, zarządzające pracą obrabiarek na podstawie programu wczytywanego na bieżąco z dziurkowanej taśmy papierowej.

Obecnie układy CNC wytwarzają dziesiątki dużych i znanych firm światowych, wśród których dominują: japoński FANUC oraz niemiecki SIEMENS. Na rynku europejskim poza firmą SIEMENS układy sterowania produkują m.in. takie firmy, jak PHILLIPS, BOSH i HEIDENHAIN. Niemal wszystkie współcześnie oferowane układy CNC podlegają wspólnemu kanonowi w zakresie struktury sprzętowej, oprogramowania oraz funkcji użytkowych.

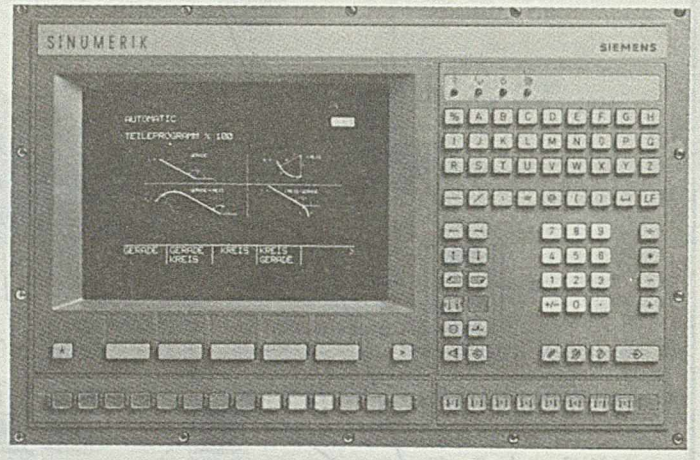
Struktura sprzętowa układów CNC stanowi zwykle oryginalne opracowanie każdej z firm, choć bywają produkowane również rozwiązania oparte na gotowych standardowych kartach (np. IBM). Odrębność tej struktury w stosunku do konfiguracji typowego komputera uniwersalnego wynika ze specjalnego przeznaczenia takiego systemu procesorowego. Specyficznym składnikiem sprzętowym układów CNC są moduły obwodów pomiarowych, spełaniające funkcję zadawania i kontroli aktualnego położenia poszczególnych osi roboczych w sterowanym obiekcie. Każdy układ CNC ma podtrzymywaną bateryjnie pamięć RAM o pojemności od kilku do kilkuset kB, służącą do przechowywania zbiorów technologicznych (programów obróbki detali dla obrabiarek lub programów sekwencji ruchów dla robotów).

Oprócz procesora głównego, układy CNC zawierają często dodatkowe postprocesory, przeznaczone np. do szybkiego wykonywania obliczeń interpolacyjnych w trakcie realizacji zaprogramowanych ruchów robota lub maszyny.

Standardowym wyposażeniem typowego CNC jest monitor monochromatyczny lub, coraz częściej, kolorowy, umożliwiający współpracę operatora z układem sterowania i obrazujący aktualne stany obiektu sterowanego oraz prowadzonego procesu obróbki bądź produkcji.

Urządzeniem wejściowym do wprowadzania przez operatora danych i inicjowania poleceń obsługi jest klawiatura (fot. 1). Zawiera ona klawisze sterujące określonymi funkcjami ruchów ręcznych obiektu, klawisze alfanumeryczne oraz klawisze funkcyjne (ang. *softkeys*).

Układy CNC są wyposażone w co najmniej jeden port transmisji szeregowej umożliwiający przyłączenie urządzeń peryferyjnych do wprowadzania i wyprowadzania tekstów zbiorów technologicznych. Integralną jednostkę w strukturze sprzętowej każdego układu CNC stanowi programowany sterownik PLC (*Programmable Logic Controller*). Zawiera on kilka wejść i wyjść, przez które układ CNC zostaje sprzęgnięty z maszyną. Sygnały wejściowe do PLC pochodzą od różnego typu czujników zainstalowanych w sterowanym obiekcie, sygnalizujących wystąpienie określonych jego stanów, np. osiągnięcie ustalonego położenia elementów ruchomych, przekroczenie progu ciśnienia itp. Wyjścia PLC sterują urządzeniami wykonawczymi w obiekcie: elektrozaworami hydraulicznymi lub pneumatycznymi, jednostkami napędowymi itp.




**Fot. 1. Widok płyty czołowej komputerowego układu sterowania numerycznego SINUMERIK 850 Produkcji firmy SIEMENS**

W zależności od przeznaczenia danego układu sterowania, w skład jego wyposażenia sprzętowego wchodzić mogą inne specjalne moduły obsługujące różnorakie funkcje.

Struktura oprogramowania układów CNC jest trójstopniowa. Trzonem a jednocześnie pierwszą warstwą oprogramowania każdego układu sterowania jest specjalizowany system operacyjny, tzw. system NC. Jest on opracowany przez producenta danego typu sterowania numerycznego i jest programem uniwersalnym działającym niezależnie od rodzaju obiektu, do którego układ CNC został zastosowany. System NC, oprócz spełniania zadań typowego komputerowego systemu operacyjnego, nadzoruje przebieg realizowanego procesu technologicznego. Stanowi on warstwę pośredniczącą między czynnikiem stymulującym pracę obiektu sterowanego (operatorem lub programem obróbki), a drugą z warstw oprogramowania – programem PLC, bezpośrednio obsługującym działanie obiektu. Kontakt pomiędzy systemem NC a programem PLC odbywa się przez wyznaczone do tego celu obszary pamięci, w której poszczególne komórki mają określone, stałe znaczenie.

Program PLC powstaje w fabryce produkującej maszyny lub roboty, w której przystosowuje się określony układ CNC do sterowania konkretnym obiektem. Program PLC jest nazwany również programem interfejsowym, ponieważ stanowi rodzaj sprzęgu (interfejsu) programowego między systemem układu sterowania a maszyną.

Do programowania sterowników PLC stosuje się specjalne języki. Zawierają one instrukcje przeznaczone do opisania operacji logicznych i arytmetycznych, rozkazy ustawiania i zerowania stanów bitów (wejściowych, wyjściowych i znaczników), rozkazy ładowania i transferu bajtów i słów, rozkazy uaktywniania i odczytywania stanów liczników i zegarów oraz rozkazy skoków.

Programowanie PLC jest specyficznym rodzajem programowania. Za pomocą rozkazów języka PLC zapisuje się w programie wszystkie możliwe sytuacje wzajemnych oddziaływań następujących składników systemu:

*OPERATOR – PROGRAM TECHNOLOGICZNY – SYSTEM NC – OBIEKT*

Działając w zamkniętym cyklu program interfejsowy na podstawie zachodzących na wejściach i wyjściach sterownika zmian stanów logicznych, wartości ustawionych uprzednio w pamięci znaczników (odpowiadających określonym stanom przejściowym procesu) oraz sygnałów na styku PLC <–> system NC, realizuje te spośród wszystkich zapisanych w programie sekwencji operacji logicznych, które w danym momencie są spełnione. Charakterystyczną cechą programu PLC jest bezpośrednie wzajemne oddziaływanie w każdej chwili poszczególnych jego fragmentów.

W niektórych typach oferowanych układów sterowania system NC zawiera gotowe podprogramy obsługujące działanie określonych typowych mechanizmów, występujących w sterowanych obiektach (np. program realizujący obsługę wrzeciona obrabiarki). Rola programu PLC w tym zakresie ogranicza się wówczas tylko do wywołania takiego podprogramu, z podaniem wartości parametrów opisujących cechy określonego obiektu rzeczywistego.

Trzeci poziom w strukturze oprogramowania komputerowych układów sterowania stanowią programy technologiczne. Powstają one u użytkowników maszyn i urządzeń sterowanych numerycznie i zawierają algorytmy realizacji określonych procesów (obróbki detali, pracy robota, montażu elementów na taśmie produkcyjnej itp.).

Programy technologiczne opracowuje się przy użyciu specjalizowanych języków programowania, obejmujących standarowy zestaw instrukcji dla typowych operacji występujących w procesach technologicznych. Ponadto technolog-programista dostaje do dyspozycji dodatkowe rozkazy, które definiuje producent maszyny, a których obsługa jest zawarta w programie PLC. Rozkazy te, umieszczone w programie technologicznym, uaktywniają działanie poszczególnych zespołów i mechanizmów danego urządzenia. Układy CNC, przeznaczone do sterowania maszynami skrawającymi, mają często zapisane w pamięci gotowe, standardowe cykle obróbcze, które mogą być wywoływane w programach technologicznych (np. cykle frezowania lub toczenia profili o zadanym kształcie.

Poza opisanymi trzema warstwami oprogramowania, istotną rolę w strukturze komputerowych układów strowania spełniają tzw. dane maszynowe. Stanowią one drugi – obok programu PLC – czynnik zapewniający dopasowanie układu CNC do obiektu, którym ma on sterować. Za pomocą danych maszynowych określa się cechy indywidualne obiektu rzeczywistego, np. liczba osi roboczych, zakresy ich ruchów, parametry napędów i układów pomiarowych itp.

Opisana struktura oprogramowania komputerowych układów sterowania numerycznego jest strukturą typową, choć nie obowiązującą dla wszystkich bez wyjątku typów sterowań. Konstruowane są również układy CNC przeznaczone do sterowania jednym określonym urządzeniem, które nie zawierają sterowania PCL: program zarządzających pracą obiektu jest wtedy częścią składową programu systemowego. Ze względu na niezmienność charakterystyki obiektu, w tym przypadku nie znajdują zastosowania również dane maszynowe. Sterowania PLC mogą być pozbawione także niektóre typy układów, charakteryzujące się rozbudowanymi możliwościami programowania technologicznego. Sterowanie działaniem mechanizmów obiektu jest wówczas realizowane bezpośrednio z poziomu programu obróbki, przejmującego funkcje programu PLC.

Realizacja każdego procesu technologicznego nadzorowanego przez układ CNC odbywa się na zasadzie wymiany informacji oraz wzajemnej współpracy zachodzącej między wyżej opisanymi składnikami systemu układów sterowania. Podział kompetencji oraz zakres działania każdej z tych warstw podczas obsługi zaprogramowanych funkcji ilustruje poniższy przykład.

W celu zrealizowania ruchu ramienia robota w osi *Z* do zadanego położenia i wzięcia przez niego do chwytaka przedmiotu, w programie technologicznym umieszczono następującą sekwencję rozkazów:

*Z*700 *M*22;

gdzie instrukcja *Z*700 oznacza pozycjonowanie osi *Z* robota do współrzędnej + 700 mm (w stosunku do punktu bazowego tej osi), natomiast instrukcja *M*22 oznacza funkcję pomocniczą realizującą zaciśnięcie szczęk łapy robota.

Program NC, po zdekowaniu w programie technologicznym powyższych instrukcji, zapala w odpowiednich komórkach pamięci sterowania (w obszarze styku system NC <–> program PLC) sygnały oznaczające informacje: *JEST ROZKAZ RUCHU W OSI Z* oraz *WYSTĄPIŁA FUNKCJA M*22. Jednocześnie kontroluje on, czy zaprogramowana wartość pozycji osi mieści się w dopuszczalnym dla niej i zdefiniowanym w danych maszynowych zakresie ruchów.

Po odczytaniu tych informacji program PLC w pierwszym rzędzie analizuje, czy spełnione są warunki realizacji ruchu ze strony samego obiektu, tzn. czy nie występują żadne stany awaryjne, czy wszytkie mechanizmy robota oraz obiektów towarzyszących znajdują się w dopuszczalnych pozycjach spoczynkowych, czy ruch osi nie spowoduje kolizji ramienia robota z którymś z pozostałych elementów systemu itd. Po sprawdzeniu powyższych warunków i uzyskaniu wymaganych potwierdzeń, program PLC wydaje do poziomu NC zezwolenie na realizację ruchu.

Sam ruch osi odbywa się pod nadzorem systemu NC. Ustawia on, odpowiedni dla zaprogramowanego kierunku i prędkości ruchu, sygnał wartości zadanej dla napędu, a podczas przejazdu osi kontroluje jej aktualne położenie na podstawie sygnałów przychodzących z układu pomiarowego. Po osiągnięciu przez oś zadanej współrzędnej (+ 700 mm), jej ruch zostaje zatrzymany, a do PLC zostaje przesałana informacja *ZGODNOŚĆ POZYCJI*. W tym momencie program PLC cofa zezwolenie dla działania regulatora osi i przystępuje do obsługi funkcji *M*22.

Kontroluje on prawidłowość poziomu ciśnienia zasilania pneumatycznych układów wykonawczych, po czym wysterowuje elektrozawór realizujący zaciskanie szczęk łapy robota. Gdy wykonanie tego rozkazu zostaje potwierdzone sygnałem czujnika (sensora zbliżeniowego lub mikrołącznika), program PLC uznaje obsługę funkcji *M*22 za zakończoną i wydaje do pozimu NC zezwolenie na wczytanie kolejnych instrukcji z programu technologicznego.

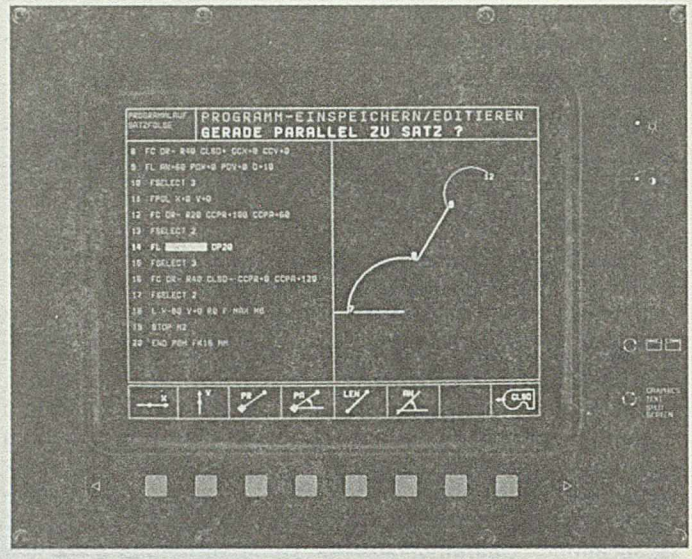
Na podobnej zasadzie odbywa się realizacja innych funkcji występujących w programach obróbki w większości typów układów sterowania numerycznego. Zadawanie i kontrola położenia osi roboczych obiektu jest jednym z podstawowych zadań, jakie mają do spełnienia układy CNC podczas wykonywania każdego programu. W przypadku sterowania robotem, przejazdy osi odpowiadają zaprogramowanym ruchom ramienia robota, natomiast w przypądku maszyn skrawających, ruchy osi realizują pozycjonowanie narzędzia względem przedmiotu obrabianego.

Jako wskaźniki pozycji rzeczywistych osi współpracujące z układami CNC mogą być stosowane układy pomiarowe pośrednie, takie jak resolwery lub optyczne przetworniki obrotowo-impulsowe (mierzące położenie kątowe wału silnika lub śruby napędowej osi), bądź też bezpośrednie, takie jak induktosyny lub liniały optyczne (podające bezpośrednio wartości przemieszczeń danej osi). Indukcyjne układy pomiarowe – resolwery i induktosyny – wychodzą powoli z użycia, gdyż zastępują je dokładniejsze i bardziej niezawodne optyczne wskaźniki położenia. Stosowane wskaźniki pozycji działają na zasadzie pomiaru metodą przyrostową, dlatego też przed rozpoczęciem właściwej czynności położenia osi, powinien być zrealizowany dla każdej z nich cykl poszukiwania znacznika bazowego układu pomiarowego (odpowiadającego zerowej współrzędnej danej osi robota lub obrabiarki). Jest to czynność, która inicjuje operator po każdoroazowym rozpoczęciu pracy obiektu.

Układy CNC, poza swoją podstawową funkcją – bezpośrednim sterowaniem przebiegiem procesów technologicznych – umożliwiają wykonywanie przez operatora wielu czynności przygotowawczych i pomocniczych związanych z realizacją danego procesu. Większość układów sterowania charakteryzuje się dużą funkcjonalnością obsługi, opartą na zasadzie zmienności obrazów odpowiadających różnym rodzajom pracy układu. Poszczególne obrazy są wywołane przez operatora przez naciskanie programowanych klawiszy funkcyjnych. Na ekranie monitora zawsze jest widoczne aktualne menu przyporządkowania klawiszy funkcyjnych dla danego obrazu.

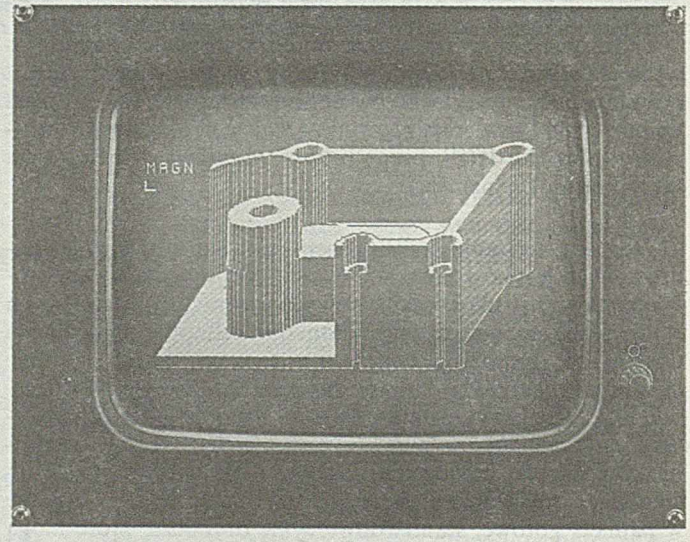
Jedną z podstawowych gałęzi w strukturze drzewa obsługi układów sterowania stanowią obrazy umożliwiające edycję zbiorów technologicznych. Użytkownik może jej dokonywać w czasie wolnym od obróbki, a także w trakcie realizacji innego procesu technologicznego. Do tego celu służą specjalne edytory pozwalające wpisywać, kasować i zmieniać funkcje występujące w programach technologicznych. Wiele typów układów CNC oferuje przy tym różnego stopnia wspomaganie programowania, polegające np. na wyświetlaniu na ekranie zarysów najczęściej spotykanych kształtów konturów, po jakich mogą poruszać się osie sterowanego obiektu, a które użytkownik może umieszczać w programie. Podaje on wówczas z klawiatury jedynie wartości współrzędnych punktów charakterystycznych danego odcinka konturu, unikając wykonywania skomplikowanych często obliczeń trajektorii (fot. 2).

Specyficznym rodzajem wspomagania operatora przy tworzeniu programów technologicznych jest tryb pracy „uczenie – odtwarzanie". Umożliwia on zapamiętywanie wybranych położeń osi obiektu – przemieszczanych np. przy pomocy posuwów ręcznych – i wprowadzanie ich do programu jako wartości zadanych dla rozkazów ruchu. Podczas realizacji programu zapamiętane sekwencje przemieszczeń osi zostają w identyczny sposób odtworzone. Powyższy tryb edycji programu znajduje zastosowanie szczególnie przy programowaniu robotów. Operator, wyznaczając doświadczalnie poszczególne fazy ruchów, może „nauczyć" ramię robota, po jakiej trajektorii ma się ono poruszać wykonując np. zadanie przenoszenia przedmiotów z miejsca na miejsce.




**Fot. 2. Edycja programu obróbki przy zastosowaniu graficznego wspomagania programowania dla układu sterowania TNC 407 produkcji firmy HEIDENHAIN**

Poza różnorodnymi formami edycji programów technologicznych z klawiatury układu, operator ma możliwość wczytywania do pamięci sterowania numerycznego gotowych programów przez złącze szeregowe, z różnego typu urządzeń zewnętrznych: czytników taśmy, pamięci dyskowych i kasetowych, a także komputerów pełniących rolę bazy danych technologicznych.




**Fot. 3. Widok obrazu symulacji graficznej obróbki przedmiotu na ekranie układu TNC 407 produkcji firmy HEIDENHAIN**

Większość układów CNC daje możliwość przeprowadzania na ekranie monitora graficznej symulacji programów technologicznych.

**dokończenie na s. 28**

# JUNISOFTEX Sp. z o.o.

44-100 GLIWICE ul. Konstytucji 11,
tel.-faks 31-75-10, 31-90-81 do 88 w. 250, 272, 282, teleks 036233

**JUNISOFTEX** – to firma z tradycjami i najdłuższymi doświadczeniami w eksploatacji wielodostępnych systemów komputerowych Novell w kraju.

**JUNISOFTEX** – to najstarszy w kraju wykonawca własnych wielodostępnych systemów pracujących w sieci NetWare firmy Novell.

**JUNISOFTEX** – to autoryzowany reseller amerykańskiej firmy Novell.

**JUNISOFTEX** – to dostawca zintegrowanych systemów komputerowych działających w kilkudziesięciu firmach obejmujących w każdym przedsiębiorstwie, niezależnie od formy własności i dziedziny gospodarki: finanse, majątek obrotowy, majątek trwały i nietrwały, sprzedaż, techniczne przygotowanie produkcji oraz kadry-płace.

**JUNISOFTEX** – to dostawca i wykonawca autoryzowanych sieci lokalnych firmy Novell.

**JUNISOFTEX** – to dostawca sprawdzonego sprzętu komputerowego renomowanych firm ALR, IBM, TEAM, DIGILAB, EPSON, MANNESMANN TALLY.

**JUNISOFTEX** – to wyłączny i autoryzowany dystrybutor komputerów i terminali firmy DIGILAB na Śląsku.

**JUNISOFTEX** – to dostawca i wykonawca okablowania lokalnych sieci komputerowych ARCNET, ETHERNET na kablach zwykłych i światłowodowych oraz na dwużyłowych kablach telefonicznych w przypadku stosowania oprogramowania NetWare ACCESS SERVER firmy Novell.

**JUNISOFTEX** – to nauczyciel, który chętnie podzieli się swoją wiedzą na organizowanych kursach w swojej szkole informatycznej – w tym nauczy Cię: podstaw informatyki, obsługi sprzętu komputerowego, eksploatacji własnych systemów informatycznych, edytorów tekstu: CHiWRITER, WordPerfect, arkuszy kalkulacyjnych QPRO v. 3.0, LOTUS 1-2-3 itp.

**JUNISOFTEX** – to doradca w zakresie komputerowych metod organizacji i eksploatacji systemów komputerowych.

**JUNISOFTEX** – to strażnik postępu i nowoczesności w Twojej firmie.

**JUNISOFTEX** – to gwarancja niezawodności, rzetelności i terminowości.

**JUNISOFTEX** – to Twój doradca i partner, któremu możesz zaufać, który Cię nigdy nie zawiedzie.

**JUNISOFTEX** – to partner, który Cię wysłucha i zawsze pomoże podjąć dobrą decyzję.

**Jeżeli chcesz**

- ☆ zastosować najnowocześniejsze systemy komputerowe oraz lokalne sieci komputerowe,
- ☆ zreorganizować, usprawnić i szybko skomputeryzować swoje przedsiębiorstwo,
- ☆ lepiej wykorzystać już istniejące w Twoim przedsiębiorstwie komputery,
- ☆ wymienić swoje niefortunnie zakupione oprogramowanie,
- ☆ pozbyć się problemów eksploatacyjnych, upadających systemów, nie działających komputerów,
- ☆ przeszkolić załogę swojego przedsiębiorstwa,
- ☆ dobrze zainwestować swoje pieniądze i podnieść rangę swojego przedsiębiorstwa,
- ☆ w przyszłości mieć dostęp do krajowej i światowej sieci komputerowej,

**TO ZGŁOŚ SIĘ DO NAS!**

0/17/91

KRYSTYNA KOLEŚNIK
Centrum Obliczeniowe
Politechnika Wrocławska

# Koncepcja symulacji obiektowej z wykorzystaniem języka Turbo Pascal 5.5

**W ostatnim czasie obserwuje się duże zainteresowanie nowym stylem programowania – programowaniem obiektowym. Niniejszy artykuł zawiera opis zastosowania metodologii obiektowej, a także programowania obiektowego do programowania symulacji.**

Dotychczas do programowania symulacji stosowano i implementowano w językach symulacyjnych metodologie ukierunkowane na symulację:
- działaniową (CSL),
- zdarzeniową (SIMSCRIPT, SIMON, PASCALSIM),
- procesową (SIMULA, LOGLAN, ADA).

Propozycję metodologii programowania symulacji ukierunkowanej obiektowo, nowego systematycznego podejścia do symulacji przedstawiono w pracach [2, 3]. Jej implementację umożliwiło dopiero rozszerzenie języka Turbo Pascal o mechanizmy programowania obiektowego (Turbo Pascal 5.5 [1, 5]).

Koncepcja symulacji obiektowej w Turbo Pascal 5.5 zakłada opracowanie kilku modułów zawierających definicje standardowych klas obiektów (zwanych dalej obiektami Turbo), specyficznych dla programowania symulacji oraz ustalenie określonych zasad ich używania w programach symulacyjnych. Poniżej zdefiniowano zawartość tych modułów, tj. klasy obiektów, ich przeznaczenie oraz metody przewidziane do działania na tych obiektach. Metodologię programowania symulacji obiektowej zilustrowano na przykładzie korzystającym z omówionych modułów.

## Koncepcja metodologii symulacji

W programach symulacyjnych, zachowanie symulowanych systemów przedstawia się opisując zachowanie występujących w nich obiektów prowadzących obsługę oraz obiektów obsługiwanych.

W zależności od czasu istnienia obiektów w modelu symulacyjnym, rozróżniamy obiekty:
- trwałe – istniejące w modelu symulacyjnym przez cały okres symulacji,
- tymczasowe – istniejące w skończonym odcinku czasu symulacji.

Ogólny model systemu zakłada, że w modelu istnieje zbiór obiektów trwałych, prowadzących wielofazową obsługę obiektów tymczasowych. Obsługa w modelu polega na odmierzaniu czasu zajętości obiektu trwałego przez obiekt tymczasowy. Liczba faz obsługi, oraz kolejność obsługi (tj. trasa obiektu tymczasowego) są określone przez obowiązujący w danym modelu system obsługi.

Wśród obiektów trwałych systemu wyróżnia się obiekty, których działanie polega wyłącznie na przydzielaniu i odbieraniu obiektom tymczasowym określonego zasobu, którym zarządza inny obiekt trwały. W dalszych rozważaniach ten typ obiektu trwałego będzie nazwany **zasobem wspólnym** i będzie reprezentowany w modelu przez atrybuty dysponującego nim obiektu trwałego.

Klasyfikacja obiektów symulowanego systemu na obiekty trwałe, obiekty tymczasowe i zasoby jest podstawowym założeniem symulacji obiektowej.

Atrybuty i działania obiektów trwałych oraz atrybuty obiektów tymczasowych, będą opisane przez klasy obiektów Turbo Pascal 5.5. Obiekty tymczasowe modelu będą przypisane do obsługujących je obiektów trwałych, przez umieszczenie ich w zbiorach kolejkowych będących atrybutami obiektu trwałego. Działania symulacyjne obiektu trwałego są podzielone na:
- akcje własne – związane z symulacją funkcjonowania obiektu,
- reakcje – związane z obsługą odebranego obiektu tymczasowego, przesłanego przez inny obiekt trwały,
- reakcje – związane ze zwolnieniem zasobu wspólnego, wykorzystywanego przez różne egzemplarze obiektów tej samej lub różnych klas.

Na akcję obiektu trwałego może składać się:
- wygenerowanie obiektu tymczasowego i określenie jego atrybutów,
- modyfikacja atrybutów obsługiwanego obiektu tymczasowego,
- zmiana stanu własnych atrybutów,
- przesłanie obiektu tymczasowego do obiektu trwałego, prowadzącego następną fazę obsługi i określoną przez system obsługi,
- identyfikacja kolejnego obiektu tymczasowego, który będzie obsługiwany przez dany obiekt trwały,
- zapalnowanie kolejnej akcji obiektu po zakończeniu obsługi inicjowanej daną akcją.

Na reakcję obiektu trwałego może składać się:
- dołączenie obiektu tymczasowego do zbioru obiektów oczekujących na obsługę przez obiekt trwały,
- modyfikacja atrybutów obsługiwanego obiektu tymczasowego,
- zmiana stanu własnych atrybutów,
- zmiana stanu zasobu wspólnego,
- identyfikacja kolejnego obiektu tymczasowego, który będzie obsługiwany przez dany obiekt trwały po zwolnieniu zasobu,
- zaplanowanie kolejnej akcji systemu wówczas, gdy istnieje

możliwość rozpoczęcia obsługi odebranego obiektu lub obiektu oczekującego na zwolniony zasób.

Zmiana stanu własnych atrybutów obiektu trwałego dotyczy atrybutów niezbędnych do prowadzenia symulacji, a także atrybutów służących do wyznaczenia miar związanych z działaniem obiektu trwałego. Należą do nich: długości kolejek obiektów tymczasowych oczekujących na obsługę przez dany obiekt trwały, współczynnik wykorzystania obiektu trwałego lub współczynnik zajętości odnoszący się do zbioru egzemplarzy obiketów danej klasy lub zasobu globalnego.

Dla obiektów tymczasowych modelu wyznacza się ich czasy pobytu w modelu symulacyjnym, czasy oczekiwania na obsługę w określonych kolejkach oraz liczbę zdarzeń określonego rodzaju.

Uwzględniając możliwości języka Turbo Pascal 5.5 proponuje się, aby stan obiektów trwałych był opisany przez atrybuty statyczne klas obiektów Turbo, a akcje i reakcje obiektów trwałych przez metody obiektu. Pewne działania składające się na akcje i reakcje obiektu mogą być opisane przez zwykłe metody i wykonywane bezpośrednio przez metody akcji i reakcji. Metody akcji i reakcji, nazywane dalej **metodami symulacyjnymi,** powinny być odpowiednio synchroniozowane w czasie.

Ponieważ definicja klasy obiektów Turbo nie zawiera części działaniowej (treści i klasy), a język nie jest wyposażony w mechanizm współprogramów, należy stworzyć mechanizm do określania kolejności wykonywania metod symulacyjnych, określonych egzemplarzy obiektów, określonych klas.

W niniejszej koncepcji proponuje się szeregowanie symulacyjnych metod obiektów na podstawie kalendarza zdarzeń, wykorzystywanego dotychczas zarówno przez symulację zdarzeniową [4] do szeregowania zdarzeń, jak i przez symulację procesową [7] do szeregowania naprzemiennie wykonywanych procesów.

Kalendarz zdarzeń jest listą opisów zdarzeń uporządkowaną chronologicznie wg czasu zdarzenia. Dla symulacji obiektowej opis każdego zdarzenia musi zidentyfikować i określić:
- rodzaj klasy obiektów trwałych,
- egzemplarz obiektu zidentyfikowanej klasy,
- metodę obiektu,
- egzemplarz obsługiwanego obiektu tymczasowego – w przypadku, gdy obiekt trwały może obsługiwać jednocześnie skończoną liczbę obiektów tymczasowych,
- czas zdarzenia.

W symulacji zdarzeniowej stosowanej w PASCALSIMIE [4] zdarzenie było tzw. zdarzeniem globalnym, tj. aktualizującym w tej samej chwili, stan wszystkich obiektów trwałych i tymczasowych systemu, na które zdarzenie oddziaływało. W symulacji obiektowej zdarzenie jest zdarzeniem elementarnym tzn. podowodującym zmianę stanu wyłącznie związanych z nim obiektów: trwałego i tymczasowego.

Program główny symulatora dowolnego systemu zawiera:
- zapowiedź korzystania z modułów symulacyjnych i modułu typów użytkownika,
- deklaracje klas obiektów trwałych systemu,
- deklaracje egzemplarzy klas obiektów trwałych,
- implementację klas obiektów trwałych,
- deklarację procedury upływu czasu,
- inicjowanie działania obiektów trwałych,
- obsługę upływu czasu,
- wyprowadzenie wyników.

```
Unit UserTypes
Type
 Elem=      {Deklaracja typu obiektu tymczasowego}
  record
   CW: integer; {Czas wejscia}
   CRF:integer; {Czas rozpoczecia fazy}
   CO: integer; {Czas oczekiwania}
  end;
 ElemPtr=^Elem; {Dekl typu wskaźn.dla ob. tymcz.}
 ClassTypes=(zrodloc,salac,kasac,raportc);
                {Deklaracja typu do ident. klas}
 ClassMethods(zrinm,zrdom,zacm,salnm,sadom,saklrm,
 sakorm,saacm, kalnm,kadom,kaklrm,kaacm,ralnm,radom);
                {Deklaracja typu do ident. metod}
end;

Program SklepObjSim;
Uses UserTypes,Gen,CalServ,Statist,Lists;
            {Deklaracje klas obiektów}
 Type
  Zrodlo =  {Definicja wzorca egzemplarzy klasy Zrodlo}
   object
    LAMBDA:INTEGER;    {Par. rozkl. czasu pom. zglosz}
    constructor Init; {Metoda inicjująca}
    dectructor Done;  {Metoda kończąca}
    procedure Action  {Metoda akcji wlasnej}
     (var Klient:ElemPtr); {ident. wygen.klienta}
   end;
   Sala =    {Definicja wzorca egzemplarzy klasy Sala}
    object
     MIZAK  :integer;  {Par rozkl. czasu fazy zakupu}
     LK     :integer;  {Liczba koszy}
     LKW    :integer;  {Liczba koszy wolnych}
     MINKASA:integer;  {Numer kasy o min. kolejce}
     KKOSZ  :List;     {Kolejka po kosze}
     DKKOSZ :Stat;     {Stat. dlug. kol. po kosze}
     ZKOSZ  :Stat;     {Stat zajętości koszy}
     constructor Init  {Metoda inicjująca}
      (i:integer)      {Liczba koszy}
     destructor Done;  {Metoda kończąca}
     procedure Min     {Proc. wyzn. numer kasy o min
                        kolejce}
      (var j:integer); {numer kasy}
     procedure KlientReaction
                       {Met. reakcji sali na we. klienta}
      (var Klient:ElemPtr);{ref do klienta wchodz.}
     procedure KoszReaction
                       {Met.reakcji sali na zwoln. kosza}
      (var Klient:ElemPtr);{ref do klienta któremu
                            przydzielono kosz}
     procedure Action {Met akcji własnej sali}
      (var Klient:ElemPtr);{ref do obsl. klienta}
    end;
   Kasa =     {Definicja wzorca egzemplarzy klasy Kasa}
    object
     NRK   :integer;   {Numer kasy}
     MIPLAC:integer;   {Par rozkl. czasu fazy plac.}
     KKASY :List;      {Kolejka do kasy}
     DKKASY:Stat;      {Stat. dlug. kolejki do kasy}
     KASY  :Stat;      {Stat.wsp. wykorzystania kas}
     constructor Init  {Metoda inicjująca}
      (i:integer);     {Numer egzemplarza}
     destructor Done   {Metoda kończąca}
      (i:integer);     {Numer egzemplarza}
     procedure KlientReaction
                       {Met. reakcji kasy na klienta}
      (var Klient:ElemPtr);
                       {ref do klienta wchodz. do kasy}
     procedure Action  {Metoda akcji wlasnej kasy}
      (var Klient:ElemPtr);{ ref do klienta obsług.}
    end;
   Raport = {Definicja wzorca egzemplarzy klasy Raport}
    object
     LKO :Counter;     {Liczba klientow obsłużonych}
     ZKAS:Stat;        {Stat. zajętości kas}

     TOB :Rand;        {Stat. czasu pobytu}
     TOCZ:Rand;        {Stat. czasu oczekiwania}
     constructor Init;{Metoda ust. stan początkowy}
     destructor Done; {Metoda wyprow. wyniki}
    end;
  var
   TKS,                {Czas konca symulacji}
   N,                  {Liczba kas}
   LKOSZY    :integer; {Liczba koszy}
   EZRODLO   :Zrodlo   {Obiekt zrodlo}
   EKASY     :array[1..10] of Kasa;
                       {Klasa obiektow Kasa}
   ESALA     :Sala     {Obiekt sala}
   ERAPORT   :Raport;  {Obiekt raport}
   EGENERATOR:Generator; {Obiekt generator}

implementation
              {Implementacja metod klas obiektów}
 constructor Zrodlo.Init;
              {Wprowadza par. strumienia zgłoszeń,
               planuje metodę akcji źródła}
  var
   Czas:longint;
  begin
   Generator.ReadPar1(expg,Lambda);
   WRITELN('PAR.ROZK.CZASU POM.ZGL.      ',LAMBDA);
   Czas:=Generator.GenExp(LAMBDA);
   _calendar.Plan(zrodloc,zracm,0,nil,Czas);
 end; {Zrodlo.Init}
```

```
destructor Zrodlo.Done;
 begin
 end; {Zrodlo.Done}

procedure Zrodlo.Action(var Klient:ElemPtr);
                {Generuje egz. klienta i planuje metodę
                 reakcji sali, planuje nast. akcję źródła}
 var
  Czas:longint;
 begin
  new(Klient);
  Klient^.CW:=calendar.Time;
  Klient^.CO:=0;
  _calendar.Plan(salac,saklrm,0,Klient,_calendar.Time);
  Czas:=_calendar.Time+G.GenExp(LAMBDA);
  _calendar.Plan(zrodloc,zracm,0,nil,Czas);
 end; {Zrodlo.Action}

constructor Sala.Init(l:integer);
                {Wprow. par. czasu obsługi przez salę
                 sprzedaży, ust. stan pocz. egzemp. sali}
 begin
  LK:=l;
  LKW:=LK;
  EGENERATOR.ReadPar1(expg,MIZAK)
  KKOSZ.Init('KKOSZ');
  DKKOSZ.Init('DKKOSZ');
  ZKOSZ.Init('ZKOSZ');
 end; {Sala.Init}
destructor Sala.Done;{Oblicza i wyprow. wyniki egz. sali}
 begin
  WRITELN('Dlugosc kolejki do koszy,);
  DKKOSZ.Done;
  WRITELN('Liczba koszy zajetych');
  ZKOSZY.Done;
 end; {Sala.Done}

procedure Sala.Min(var j:integer;);
                {Wyznacza numer kasy o min. kolejce}
 Var l:integer;
 wmin:integer;
 begin
  j:=1;
  wmin:=EKASY[j].KKASY.ListSize;
  for l:=2 to n DO
   if EKASY.KKASY.ListSize < wmin
    then       begin
           j:=l;
           wmin:=EKASY.KKASY.ListSize;
          end;
 end; {Sala.Min}

procedure Sala.KlientReaction(Klient:ElemPtr);
                {Reaguje na wejście klienta, planuje akcję
                 sali jeżeli są wolne kosze lub dołącza
                 klienta do kolejki oczekujących na kosze }
 var Czas:longint;
 begin
  if  LKW<>0
   then begin
         LKW:=LKW-1;
         ZKOSZ.Act(1);
         Czas:=EGENERATOR.GenExp(MIZAK);
         _calendar.Plan(salac,saacm,0,Klient,Czas);
        end
   else begin
         KKOSZ.AddLast(Klient);
         DKKOSZ.Act(1);
         Klient.CRF:=_calendar.Time;
        end;
 end; {Sala.KlientReaction}

procedure Sala.KoszReaction(Klient:ElemPtr);
                {Reaguje na zwolnienie kosza, planuje akcję
                 sali, gdy jest kolejka po kosze, akt. stan
                 zasobu koszy}
 begin
  if KKOSZY.ListSize<>0
   then begin
         KKOSZY.TakeFirst(Klient);
         DKKOSZ.Act(-1);
         Klient.CO:=_calendar.Time-Klient.CRF;
         Czas:=EGENERATOR.GenExp(MIZAK);
         _calendar.Plan(salac,saacm,0,Klient,Czas);
        end
   else begin
         LKW:=LKW-1;
         Klient:=nil;
         ZKOSZ.Act(-1);
        end;
 end; {Sala.KoszReaction}

procedure Sala.Action(var Klient:ElemPtr);
                {Przesyła klienta do wybranej kasy -
                 planuje metodę reakcji kasy na klienta}
 begin
  Min(MINKASA);
  _calendar.Plan(kasac,kaklrm,MINKASA,Klient,_calendar.Time);
 end; {Sala.Action}

constructor Kasa.Init(l:integer);
                {Wprowadza par. czasu obsługi w kasie,
                 ustala stan pocz. egzempl. kasy}
var
 begin
  NRK:=l;
  EGENERATOR.ReadPar1(expg,MIPLAC);
  KKASY.Init;
  DKKASY.Init('DKKASY');
  WIKASY.Init('WIKASY');
 end; {Kasa.Init}

destructor Kasa.Done;
               {Oblicza i wyprowadza wyniki egzempl. kasy}
```

Na obsługę upływu czasu wykonywaną podczas trwania przebiegu symulacji składa się:
- pobieranie z kalendarza opisu najbliższego zdarzenia,
- przesunięcie czasu symulacji do czasu tego zdarzenia,
- identyfikacja obiektu trwałego i jego metody,
- wykonanie tej metody w celu obsługi zdarzenia.

Deklaracje klas obiektów trwałych w programach symulacyjnych wykorzystują zdefiniowane standardowe obiekty Turbo i ich metody, zawarte w modułach stanowiących uzgodnienia symulacyjne.

W każdym programie symulacyjnym oprócz klasy reprezentujących elementy symulowanego systemu, muszą być zdefiniowane dwie dodatkowe klasy obiektów trwałych. Są to: klasa obiektów reprezentująca źródło zgłoszeń po obsługę oraz klasa reprezentująca obiekt *Raport* zbierający statystyki obiektów tymczasowych oraz statystyki dotyczące całych klas obiektów trwałych. Podział działań obiektu na metody akcji i reakcji ogranicza dostęp zdalny do atrybutów innego obiektu trwałego tylko do odczytu, a tym samym każdy obiekt trwały może modyfikować tylko własne atrybuty. Wyjątek stanowi obiekt *Raport*, którego atrybuty statystyczne mogą być aktualizowane przez egzemplarze innych obiektów.

## Udogodnienie symulacyjne

Udogodnienia symulacyjne, ułatwiające programowanie standardowych funkcji symulacyjnych, będą podzielone na moduły klas obiektów Turbo, uwzględniające ich zastosowania. Zdefiniowane klasy służą do:
- obsługi kolejkowych zbiorów obiektów trwałych,
- wyznaczania miar pracy symulowanego systemu,
- generowania liczb pseudolosowych,
- obsługi kalendarza zdarzeń.

Pełna specyfikacja modułów została podana w [5].

Wymaga się, aby użytkownik modułów sam napisał moduł o standardowej nazwie *UserTypes* definiujący atrybuty obiektów tymczasowych symulowanego systemu, wykorzystywany przez moduły standardowe.

## Moduł obiektów listowych – Lists

Obiekty zadeklarowane w module służą do obsługi kolejkowych zbiorów obiektów tymczasowych. W module *Lists* zadeklarowano standardowy typ elementu listy oraz klasę nagłówka listy. Zdefiniowano w niej metody realizujące podstawowe operacje na listach. Są to:
- inicjalizacja nagłówka listy – *Init*,
- zwolnienie listy obiektów tymczasowych z kasowaniem elementów listy i obiektów tymczasowych – *Done*,
- podanie referencji do obiektu tymczasowego z pierwszego elementu listy – *FirstElem*,
- podanie liczby elementów listy – *ListSize*,
- dołączenie na koniec listy elementu wskazującego na dołączany obiekt tymczasowy – *AddLast*,
- usunięcie elementu z początku listy z podaniem odczytanej referencji do usuwanego obiektu tymczasowego – *TakeFirst*.

## Moduł obiektów statystycznych – Statist

Moduł obiektów statystycznych *Statist* służy do ułatwienia wyznaczania miar pracy symulowanego systemu. Do wyznaczanych w procesie symulacji miar należą:
- liczniki systemowe,
- liczniki zdarzeń określonego rodzaju dla danego obiektu,

– czasy przejścia,
– współczynniki wykorzystania,
– współczynniki zajętości.

Wyznaczanymi parametrami miar są uogólnione parametry statystyczne: wartość średnia, odchylenie standardowe, wartość minimalna, wartość maksymalna, wielkość próby oraz histogram.

W module zadeklarowano pięć klas obiektów statystycznych. Ponieważ niektóre klasy zawierają elementy wspólne, zdefiniowano hierarchie obiektów dziedziczących cechy klas wcześniej zadeklarowanych.

Miary typu „liczniki" są wyznaczane dla całego systemu, opierając się na klasie *Counter*.

Miary typu „zmienna losowa", tj. czas przejścia lub liczba zdarzeń określonego rodzaju, są wyznaczane dla obiektów tymczasowych, opierając się na klasach *Rand* i *RandHs*.

Miary typu „proces stacjonarny", tj. współczynniki wykorzystania i współczynniki zajętości, są wyznaczane dla obiektów trwałych, opierając się na klasach *Stat* i *StatHs*.

W module zadeklarowno następujące stałe i klasy:
- stałe określające maksymalny wymiar histogramu dla obiektów klasy *RandHs – RANDHIS* oraz dla obiektów klasy *SatatHs – STATHIS*,
- klasę obiektów *Rand*, wyznaczającą zwykłą średnią oraz zwykłe odchylenie standardowe,
- klasę obiektów *RandHs*, wyznaczającą zwykłą średnią, zwykłe odchylenie standardowe oraz histogram,
- klasę obiektów *Stat*, wyznaczającą ważoną średnią oraz ważone odchlenie standardowe,
- klasę obiektów *StatHs*, wyznaczającą ważoną średnią, ważone odchylenie standardowe oraz histogram,
- klasę obiektów *Counter*, zliczającą liczbę zdarzeń określonego rodzaju.

Atrybuty statyczne tych klas są związane z wyznaczanymi parametrami statystycznymi miar, zaś metody służą do:
– ustalenia stanu początkowego atrybutów statystycznych oraz histogramu – *Init*,
– wyznaczenia i wyprowadzenia uogólnionych parametrów statystycznych oraz opcjonalnie histogramu – *Done*,
– aktualizacji stanu atrybutów statycznych oraz histogramu wg wartości aktualnej – *Act*,
– wyznaczenia uogólnionych parametrów statystyczych – *Caculate*,
– wyprowadzenia uogólnionych parametrów statystycznych i histogramu w trybie tekstowym, w określonym formacie na aktualne wyjście – *TextOut*.

## Moduł obiektów do generacji liczb pseudolosowych

Moduł *Gen* służy w programach symulacyjnych do generowania liczb pseudolosowych i umożliwia generowanie liczb wg ośmiu rozkładów. W modelu zadeklarowano następujące stałe i typy:
- stałą określającą maksymalny wymiar tablicy wartości oraz tablicy prawdopodobieństw dla rozkładu dysktretnego – *MaxPartNum*,
- typ na tablicę wartości dla rozkładu dyskretnego – *ValArr*,
- typ na tablicę prawdopodobieństw dla rozkładu dyskretnego – *ProArr*,

```
begin
 WRITELN('Kolejka    NR    ',NRK:3);
 WRITELN('Dlugosc');
 DKKASY.Done;
 WRITELN('Wspolczynnik wykorzystania ');
 WKASY.Done;
end; {Kasa.Done}

procedure Kasa.KlientReaction(Klient:ElemPtr);
                {Reaguje na przybycie klienta - dołącza
                 klienta do kolejki, a gdy kasa jest wolna
                 planuje metodę akcji własnej}
 var
  Czas:longint;
 begin
  KKASY.AddLast(Klient);
  DKKAS.Act(1);
  Klient^.CRF:=_calendar.Time;
  if  KKASY.ListSize=1
   then begin
          ERaport.ZKAS.Act(1);
          Czas:=Klient^.CO;
          ERaport.TOCZ.Act(Czas);
          WKASY.Act(1);
          Czas:=_calendar.Time+EGENERATOR.GenExp(MIPLAC);
          _calendar.Plan(kasac,kaacm,NRK,Klient,Czas);
        end;
 end; {Kasa.KlientReaction}

procedure Kasa.Action(var Klient:ElemPtr);
                {Oblicza miary dla klienta, planuje metodę
                 reakcji sali na zwolnienie kosza,
                 planuje następną akcję kasy}
 var
  Czas:longint;
 begin
  Klient:=KKASY.TakeFirst;
  DKKASY.Act(-1);
  Czas:= _calendar.Time-Klient^.CW;
  ERaport.TOB.Act(Czas);
  ERaport.LKO.IncCounter;
  dispose(Klient);
  _calendar.Plan(salac,sakorm,0,nil,_calendar.Time);
  if KKASY.ListSize<>0
   then begin
          Klient:=KKASY.First;
          Czas:=Klient^.CO+_calendar.Time-Klient^.CRF;
          ERaport.TOCZ.Act(Czas);
          Czas:=_calendar.Time+EGENERATOR.GenExp(MIPLAC);
          _calendar.Plan(kasac,kaacm,NRK,Klient,Czas);
        end;
   else begin
          WKASY.Act(-1);
          ERaport.ZKAS.Act(-1);
        end;
 end; {Kasa.Action}

constructor Raport.Init;
                {Inicjuje statystyki globalne, inicjuje
                 egzemplarze obiektów}
 begin
  EGENERATOR.Init;
  _calendar.Init;
  LKO.Init('LKO');
  ZKAS.Init('ZKAS');
  TOB.Init('TOB');
  TOCZ.Init('TOCZ');
  writeln('Podaj liczbe kas');
  readln(N);
  writeln('Podaj liczbe koszy');
  readln(LKOSZY);
  writeln('Podaj czas symulacji');
  readln(TKS);
  writeln('Podaj parametr dla generatora zgloszen');
  EZRODLO.Init;
  writeln('Podaj parametr dla generatora czasu zakupu');
  ESALA.Init(LKOSZY);
  for i:=1 to N do
   begin
    writeln('Podaj par dla generatora czasu placenia');
    EKASY[i].Init(i);
   end;
 end; {Raport.Init}

destructor Raport.Done;
                {Oblicza miary globalne, uruchamia metody
                 wyprowadzania wyników dla egzempl. klas}
 var
  i:integer;
 begin
  writeln('Symulacja sklepu samoobslugowego');
  writeln('Wyniki');
  writeln('Liczba klientow obsłużonych ',LKO.Done;
  writeln('Czas pobytu klienta w sklepie');
  TOB.Done;
  writeln('Czas oczekiwania');
  TOCZ.Done;
  writeln('Dlugosci kolejek do kas');
  for i:=1 to N do
   EKasy[i].Done;
  writeln('Liczba kas zajetych');
  ZKAS.Done;
 end; {Raport.Done}

procedure Scan1;
  begin
  end;
{$I c:\objslm\ScanFile.pas}
begin
 ERAPORT.Init;
 while _calendar.Time<TKS do Scan;
 ERAPORT.Done;
end.
```

**Na wydruku (strony 15–17) zamieszczono treść modułu UserTypes oraz programu symulacyjnego**

● typ wyliczeniowy z wartościami oznaczającymi typy generatorów,
oraz klasę obiektów *Generator* z metodami wykorzystywanymi do wprowadzenia parametrów i generowania liczb pseudolosowych o następujących rozkładach:
● rozkładzie równomiernym niestandardowym – *GenRand*,
● rozkładzie nierównomiernym o wartościach z przedziału (0...*n*) – *GenDiscr0*,
● rozkładzie nierównomiernym o wartościach podanych w tablicy – *GenDiscr*1,
● rozkładzie Bernoulliego – *GenBern*,
● rozkładzie normalnym – *GenNorm*,
● rozkładzie wykładniczym – *GenExp*,
● rozkładzie Erlanga – *GenErl*,
● rozkładzie Poissona – *GenPoiss*,
a także metodami do obsługi komunikacji z użytkownikiem w celu wprowadzenia parametrów generatora – *ReadPar*1, *ReadPar*2, *ReadPar*3.

## Moduł obiektów do obsługi kalendarza zdarzeń CalServ

Moduł do obsługi kalendarza zawiera deklaracje służące do organizowania kalendarza zdarzeń oraz metody związane z jego uaktualnianiem.

Metoda realizująca upław czasu składa się z części standardowej, która pobiera opis zdarzenia z kalendarza i przesuwa czas symulacji (procedura *Scan*), oraz z części unikalnej, która identyfikuje klasę obiektów, egzemplarz i metodę (procedura *Scan*1) oraz wywołuje metodę symulacyjną z określonymi parametrami.

Metoda *Scan*1 powinna być napisana przez użytkownika modułów symulacyjnych, ponieważ wymaga znajomości nazw metod symulacyjnych zdefiniowanych w deklaracjach jego klas, opisujących działania obiektów trwałych.

Ze względu na wymaganą kolejność modułów standardowych, wynikającą z używania standardowych metod standardowych klas oraz metod symulacyjnych w deklaracjach klas użytkownika, obydwie metody, związane z upływem czasu, nie mogą być zamieszczone w module obsługi kalendarza, natomiast muszą być włączane do programu głównego po implementacji metod klas użytkownika. Metoda standardowa *Scan* będzie włączana w postaci źródłowej z pliku *ScanFile*.

Proponuje się, aby metoda *Scan*1 wykorzystywała standardowy algorytm identyfikacji klasy i metody na podstawie identyfikatora klasy i metody, zawartego w opisie zdarzenia pobranego z kalendarza. Identyfikatorami klas i metody będą wartości dwóch typów wyliczeniowych zadeklarowane przez użytkownika, także w module *UserTypes*, reprezentujące w pierwszym z nich klasy, a w drugim – metody obiektów trwałych.

W module *CalServ* zadeklarowano standardowe typy, klasy obiektów i egzemplarze, a mianowicie:
– rekord opisu zdarzenia – *EvNotice*,
– typ wskazujący na opis zdarzenia – *EvPtr*,
– klasę obiektu nagłówkowego kalendarza zdarzeń – *CalHeader*,
– egzemplarz nagłówka kalendarza,
– bieżący czas symulacji.

Metody działania na kalendarzu zdarzeń umożliwiają:
● inicjalizację nagłówka kalendarza – *Init*,
● zwolnienie listy kalendarza – *Done*,
● podanie bieżącego czasu symulacji – *Time*,
● wstawienie opisu zdarzenia do kalendarza – *Plan*,
● usunięcie opisu zdarzenia z kalendarza – *DelEv*,
● podanie czasu pierwszego zdarzenia – *FirstEvTime*,
● podanie klasy pierwszego zdarzenia – *FirstEvClass*,
● podanie metody pierwszego zdarzenia – *FirtEvMeth*,
● podanie numeru egzemplarza obiektu trwałego pierwszego zdarzenia – *FirstEvEntIdent*.

W wymienionych powyżej deklaracjach wykorzystuje się niestandardowe typy użytkownika o standardowych nazwach, tj. *ClassTypes*, *ClassMethods*, *Elem*, *ElemPtr* – zdefiniowane w module *UserTypes*.

## Przykład programu symulacyjnego zorientowanego obiektowo

Jako przykład ilustrujący metodę symulacji zorientowanej na obiekty, a wykorzystujący omówione powyżej standardowe klasy, przyjęto model sklepu samoobsługowego.

W modelu tym zakłada się istnienie obiektów trwałych, reprezentowanych przez egzemplarze klas obiektów. Są to:
● źródło zgłoszeń klientów *EZRODLO* – egzemplarz obiektu klasy *Źródło*,
● sala sprzedaży *ESALA* – egzemplarz obiektu klasy *Sala*,
● określona liczba kas *EKASY* – egzemplarze obiektów klasy *Kasa* (liczba egzemplarzy określana przez czynnik *n*),
● Raport – umowny obiekt trwały do zbierania statystyk dotyczących obiektów tymczasowych i klas obejmujących więcej niż jeden egzemplarz obiektu.

Tymczasowym obiektem obsługiwanym jest klient, reprezentowany przez egzemplarz typu *Elem*. Na wydruku zamieszczono treść modułu *UserTypes* oraz programu symulacyjnego.

* * *

Metodologia symulacji obiektowej jest systematyczną metodą budowania modeli symulacyjnych. W połączeniu z programowaniem obiektowym i opracowanymi modułami do zastosowań symulacyjnych ułatwi ona programowanie poprawnych modeli symulacyjnych systemów rzeczywistych. Omówiona koncepcja została zaimpelentowana wraz z dodatkowymi udogodnieniami w postaci modułów przeznaczonych do śledzenia stanu modelu podczas wykonywania przebiegu symulacji, składowania wyników symulacji w pliku oraz prezentacji wyników symulacji w postaci graficznej.

Metoda symulacji obiektowej daje nieco dłuższe kody programów niż symulacja zdarzeniowa (przy założeniu, że dla obydwu metod są dostępne biblioteki standarowych procedur symulacyjnych), gdyż działania jednego zdarzenia są rozbite na kilka metod. Programowanie modelu jest jednak wykonywane w sposób systematyczny, co łatwiej pozwala napisać poprawny program modelu.

Planuje się prowadzenie dalszych prac mających na celu wprowadzenie nowych udogodnień do programowania symulacji. Będą nimi kolejno:
● przejście na stosowanie w standardowych modułach wyłącznie programowania obiektowego (w obecnej wersji obiektowo są programowane działania obiektów trwałych, w kolejnej implementacji przewiduje się obiektowy dostęp także do atrybutów obiektów tymczasowych),

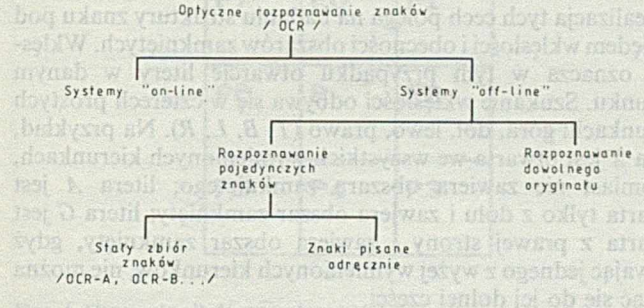
**dokończenie na s. 24**

IWONA DĄBROWSKA
DOROTA DEPOWSKA
Instytut Elektrotechniki Teoretycznej
i Miernictwa Elektrycznego
Politechnika Warszawska

# Metody automatycznego rozpoznawania znaków

Automatyczne rozpoznawanie znaków, nazywane w skrócie OCR (ang. *Optical Character Recognition*), jest interesującą i dynamicznie w ostatnich latach rozwijaną gałęzią szerszej dziedziny analizy obrazów.

Automatyczne systemy rozpoznawania pisma dzielą się na dwie kategorie w zależności od trybu wprowadzania danych. Pierwszą grupą są systemy typu on-line, dla których danymi wejściowymi są znaki pisane odręcznie i wprowadzone za pomocą specjalnej elektronicznej tablicy cyfrowej. Wprowadzane znaki są na bieżąco analizowane i rozpoznawane. W procesie przetwarzania może być więc brany pod uwagę element czasowy jako jedna z informacji o każdym fragmencie znaku. Drugą kategorią systemów OCR są systemy typu off-line, używające do wprowadzania znaków urządzeń optycznych (rys. 1) [11]. Systemy typu on-line, wymagające specjalnej aparatury (tablic elektronicznych), nie będą tu szerzej omawiane.

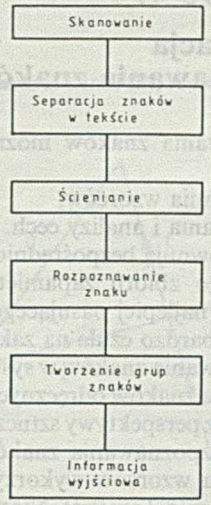



**Rys. 1. Optyczne rozpoznawanie znaków**

W systemach off-line tekst podlegający rozpoznaniu jest wprowadzany do komputera za pomocą np. skanera optycznego i umieszczany w specjalnym zbiorze, z którego korzysta program rozpoznający.

Rozpoznawanie typu off-line można z kolei podzielić na dwie grupy: rozpoznawanie pojedynczych znaków oraz rozpoznawania dowolnego oryginału. W drugiej grupie mieszczą się między innymi badania związane z identyfikacją podpisów, gdzie podpis traktuje się jako całość porównywaną ze wzorcem.

Najszerszą grupę badań OCR można ogólnie nazwać rozpoznawanie pojedynczego znaku. W tej kategorii mieszczą się zarówno systemy rozpoznawania identyfikujące znaki drukowane jedną czcionką (systemy jednozbiorowe), systemy identyfikujące znaki drukowane kilkoma (wieloma) czcionkami (systemy wielozbiorowe), jak i systemy identyfikujące odseparowane znaki pisane odręcznie. Rys. 2 przedstawia ogólny schemat blokowy systemu rozpoznającego pojedyncze znaki.




**Rys. 2. Schemat przykładowego systemu rozpoznawania znaków**

**IWONA DĄBROWSKA** ukończyła w 1990 r. Wydział Elektryczny Politechniki Warszawskiej i została przyjęta do Studium Doktoranckiego PW. Obecnie prowadzi badania nad metodami rozpoznawania pisma pod względem ich przydatności do rozpoznawania pisma ręcznego.

**DOROTA DEPOWSKA** ukończyła w 1990 r. Wydział Elektryczny Politechniki Warszawskiej i została przyjęta do Studium Doktoranckiego PW. Obecnie prowadzi badania nad metodami rozpoznawania pisma pod względem ich przydatności do rozpoznawania pisma ręcznego.

## Przetwarzanie wstępne

Jak już wspomniano, tekst przeznaczony do automatycznego rozpoznawania jest wprowadzany za pomocą skanera, zazwyczaj całymi stronami, rzadziej wierszami (skanery ręczne nie są polecane do tych celów, jako wprowadzające zbyt wielki błąd).

Pierwszą czynnością, wykonywaną przez system rozpoznawania, jest odseparowanie poszczególnych znaków w tekście. Dokonuje się tego stopniowo, tzn. najpierw dzieli się stronę tekstu na wiersze (zakładając, że strona nie zawiera grafiki lub grafika już została wydzielona), następnie wiersze dzieli się na poszczególne słowa, a w słowach separuje się znaki.

Na tym etapie może być dokonana wstępna klasyfikacja znaków [3] – w zależności od wysokości znaku w słowie – na następujące grupy:
- znaki wysokie, np. *t, d, h;*
- znaki centralne, np. *a, c, s;*
- znaki niskie, np. *p, q, g;*
- znaki pełne, np. *j;*
- znaki niepełne:
  - środkowe, np. =, +;
  - podwyższone, np. ', *;
  - obniżone, np., .

Ta wstępna klasyfikacja jest przydatna zwłaszcza w systemach rozpoznających wielozbiorowych, ponieważ od razu na wstępie drastycznie zmniejsza liczbę wzorców, z którymi należy porównać rozpoznany znak. Klasyfikowanie odbywa się na podstawie porównania wysokości słów i pojedynczych liter i kończy się nadaniem każdemu znakowi etykiety określającej przynależność do danej grupy.

## Ogólna klasyfikacja technik rozpoznawania znaków

Techniki rozpoznawania znaków można sklasyfikować w dwóch grupach jako:
a) techniki porównywania wzorców,
b) techniki porównywania i analizy cech.
Metody grupy (a) porównują bezpośrednio znak wejściowy do znaków standardowego zbioru zapamiętanych prototypów. Znalezienie prototypu najlepiej pasującego kończy rozpoznawanie. Metoda ta jest bardzo czuła na zakłócenia i nie bardzo nadaje się do rozpoznawania znaków w systemach wielozbiorowych, a tym bardziej dla znaków odręcznych. Co więcej, nie jest ona przydatna patrząc z perspektywy sztucznej inteligencji, jako nie oddająca istoty rozpoznawania znaków przez człowieka. Metoda porównywania wzorców wykorzystana została m.in. w komercyjnym programie *TextPert*. Niestety program ten nie radzi sobie zupełnie z tekstami pisanymi w języku polskim, mimo, że teoretycznie zawiera wbudowane wzorce polskich liter. Należy dodać, że osiągany procent poprawnie rozpoznanych znaków w dużym stopniu zależy od rodzaju użytej czcionki.

Techniki grupy (b) są oparte na porównywaniu cech wyodrębnionych z bitowych map znaków. Wyszukiwane są najbardziej znaczące cechy znaku i porównywane z opisem cech znaku idealnego. Rozpoznanie kończy opis, który pasuje najściślej do znaku idelanego. Techniki te znacznie lepiej oddają proces wnioskowania ludzkiego, dlatego też na ich rozwój kładzie się znacznie większy nacisk.

W zależności od sposobu ekstrakcji cech, używane techniki analizy można podzielić na:
- transformacje globalne,
- cechy otrzymywane ze statystycznej dystrybucji punktów,
- cechy geometryczne i topologiczne.

Przy wykorzystywaniu metod ekstrakcji cech bardzo często lepiej i wygodniej jest posłużyć się samym szkieletem litery, a nie jej pełnym obrazem. Powoduje to znaczne zmniejszenie liczby przetwarzanych danych oraz uwypuklenie niektórych cech litery. Obecnie istnieje wiele bardzo dobrych i szybkich algorytmów ścieniających, jak np. [9].

## Wybrane metody rozpoznawania znaków

Poniżej zostaną omówione metody rozpoznawnia znaków, wykorzystujące różne zestawy cech identyfikacyjnych. Każdy znak po procesie skanowania rastrowego jest zakłócony szumami. Może on być wydłużony, mieć oddzielne punkty lub sztuczne dziury, albo też nierówne krawędzie. Eliminacja tych zakłóceń stanowi odrębne zagadnienie, które w tym artykule nie będzie omawiane.

Metody rozpoznawania będą odniesione do identyfikacji znaku po eliminacji zakłóceń oraz po ścienieniu. Kolejne znaki są umieszczone w dwuwymiarowej tablicy roboczej, stanowiącej tzw. pole robocze znaku (litery czy też cyfry). Liczba kolumn tej tablicy odpowiada szerokości znaku, zaś liczba wierszy jego wysokości (znak zostaje umieszczony w możliwie najmniejszej dla niego tablicy). Pojedynczy element tablicy odpowiada pikslowi znaku (1) lub jego tła (0). Nie rozpatruje się stanów pośrednich (stanów szarości) między kolorem czarnym i białym.

Metoda pierwsza wykorzystuje dwa zbiory cech [13]: cechy podstawowe i dodatkowe.

### Cechy podstawowe

Realizacja tych cech polega na badaniu struktury znaku pod względem wklęsłości i obecności obszarów zamkniętych. Wklęsłość oznacza w tym przypadku otwarcie litery w danym kierunku. Szukanie wklęsłości odbywa się w czterech prostych kierunkach: góra, dół, lewo, prawo (*T, B, L, R*). Na przykład, litera *X* jest otwarta we wszystkich wymienionych kierunkach, natomiast nie zawiera obszaru zamkniętego; litera *A* jest otwarta tylko z dołu i zawiera obszar zamknięty; litera *G* jest otwarta z prawej strony i zawiera obszar zamknięty, gdyż używając jednego z wyżej wymienionych kierunków, nie można dostać się do jej dolnej części.

| L | R | B | T | O | |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 | A |
| 0 | 0 | 0 | 0 | 1 | BDOPQ |
| 0 | 1 | 0 | 0 | 0 | CEF |
| 0 | 1 | 0 | 0 | 1 | G |
| 0 | 0 | 1 | 1 | 0 | HMNW |
| 1 | 1 | 0 | 0 | 0 | ISZ |
| 1 | 0 | 0 | 1 | 0 | J |
| 0 | 1 | 1 | 1 | 0 | K |
| 0 | 0 | 0 | 0 | 0 | LT |
| 0 | 1 | 1 | 0 | 1 | R |
| 0 | 0 | 0 | 1 | 0 | UVY |
| 1 | 1 | 1 | 1 | 0 | X |

**Rys. 3. Model pisma ręcznego wraz z cechami charakterystycznymi:**
**L – kierunek lewo**
**R – kierunek prawo**
**B – kierunek dół**
**T – kierunek góra**
**O – obecność obszarów zamkniętych**

Na rys. 3 zaprezentowano model pisma ręcznego wraz z jego charakterystycznymi cechami. Przy innym wyglądzie liter skład otrzymanych grup pierwotnych liter będzie inny. Należy zaznaczyć, że można jeszcze wprowadzić dodatkową cechę podstawową, a mianowicie możliwość rozróżnienia obszaru zamkniętego, jak np. w przypadku litery *O* i obszaru pseudo-zamkniętego, np. dla litery *G* (rys. 3), a także obliczać liczbę obszarów zamkniętych dla danego znaku (prosta identyfikacja litery *B* w danej grupie).

## Cechy dodatkowe

Cechy te umożliwiają odszukanie właściwej litery w grupie liter wyznaczonych przez cechy podstawowe. Cechami dodatkowymi są profil i łuk.

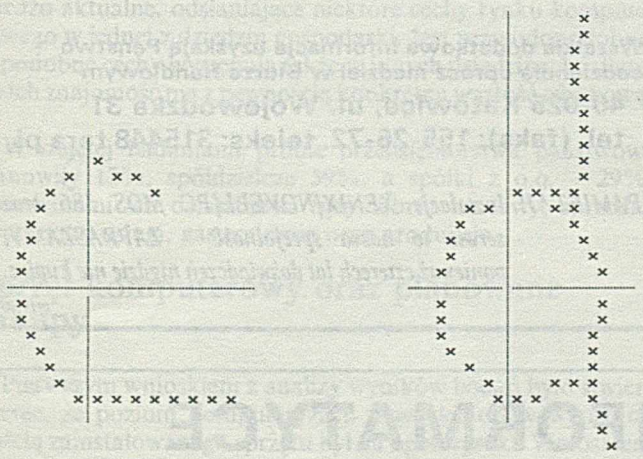
Analiza profilu jest dokonywana przez rzuty pionowe i poziome znaku. Umożliwia to m.in. rozróżnienie liter *M* oraz *W*.

Analiza kątowa polega na badaniu obecności łuku w czterech rogach obrazu litery. Na przykłd litera *O* ma tak rozumiany łuk w czterech rogach, natomiast litera *D* – tylko w dwóch rogach: prawym górnym i dolnym. Na tej podstawie można rozróżnić obie litery.

Metoda druga [10], opiera się na statystycznej analizie cech danego znaku. Wykorzystuje ona następujące cechy:
- liczbę pętli,
- liczbę *T*–połączeń; *T*–połączenie występują wtedy, gdy piksel centralny *P* (rys. 4) ma wartość logiczną jeden tzn. jest czarny, a trzy spośród ośmiu piksli jego otoczenia są czarne;
- liczbę *X*–połączeń; *X*–połączenie występuje wtedy, gdy piksel centralny *P* (rys. 4) jest czarny, a cztery (lub więcej) piksle z jego sąsiedztwa są czarne;

| | | |
|---|---|---|
| P7 | P0 | P1 |
| P6 | P | P2 |
| P5 | P4 | P3 |

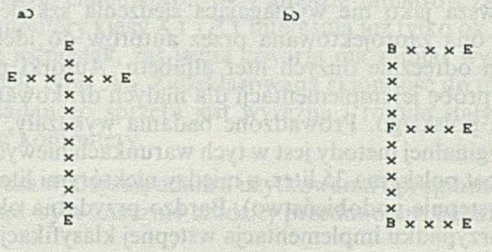
Rys. 4. Otoczenie piksla centralnego



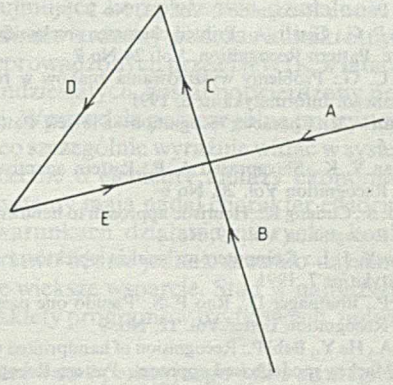
Rys. 5. Obrazy liter „c" oraz „d" wraz z osiami pionowymi i poziomymi; litera „d" ma dwa pionowe i dwa poziome przecięcia, zaś litera „c" dwa pionowe i jedno poziome przecięcie

- stosunek wysokości do szerokości znaku; na przykład takie litery, jak *j* oraz *i* mają dużą wartość tego stosunku, podczas gdy litery takie jak *m* czy *w* – małą;
- izolowane kropki; kropki te mają litery *i, j, ń, ć,* itp.
- zerowe przecięcia pionowe i poziome; przecięcia te ustala się na podstawie liczby przecięć obrazu litery odpowiednio z linią pionową lub poziomą (rys. 5); Linię tę prowadzi się przez środek ciężkości znaku, obliczony na podstawie jego momentów centralnych;
- punkty końcowe; punkt centralny *P* (rys. 4) jest punktem końcowym wtedy, gdy jest czarny i tylko jeden spośród ośmiu jego sąsiadów jest też czarny;
- aproksymujący półkrąg w kierunkach: Pn, Pd, Wsch, Zach; gdy obraz litery zostanie podzielony osią pionową (wg środka ciężkości) to litera *a* ma półokrąg w połowie zachodniej, zaś litera *b* – we wschodniej. Przy podziale poziomym litera *m* ma dwa półokręgi w połowie północnej, zaś litera *w* w południowej.

Metoda trzecia opiera się na następujących czterech cechach [7]:
- obecność punktów końca,
- obecność punktów rozwidlenia,
- obecność punktów krzyżowych,
- obecność punktów załamania.

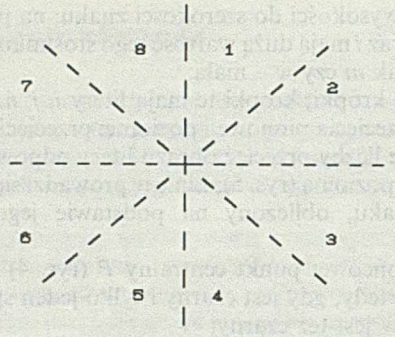



Rys. 6. Obrazy liter
a) „t"
b) „E"
wraz z oznaczeniem punktów: końca (E), rozwidlenia (F), krzyżowych (C) oraz załamania (B)




Rys. 7. Przykłady cech metody trzeciej:
dwa punkty końcowe
brak punktów rozwidlenia
jeden punkt krzyżowy
dwa punkty załamania
pięć podsegmentów etykietowanych A, B, C, D, E

Objaśnienie tych cech przedstawiono na rys. 6. Wykorzystuje się je przy podziale znaku na odpowiednie segmenty. Segmentom tym przypisuje się etykiety (rys. 8). Dla każdego segmentu odnajduje się i zapamiętuje punkty końcowe i początkowe. Oprócz tego zapamiętuje się długość danego segmentu w stosunku do długości całego znaku oraz orientację segmentu. W celu określenia orientacji wykorzystuje się w tym przypadku osiem sektorów po 45 deg. (rys. 7).




Rys. 8. Orientacje segmentów

* * *

Przedstawione powyżej metody wybrano tak, aby zaprezentować możliwie różne podejścia do zagadnienia rozpoznawania liter. Metody pierwsza oraz trzecia wykorzystują tylko zależności topologiczne i geometryczne, jednakże istotnie różnią się pod względem doboru badanych cech. Metoda druga jest przykładem statystycznego podejścia do omawianego zagadnienia.

Stosunkowo najłatwiejszą do implementacji wydaje się metoda pierwsza jako nie wymagająca śledzenia szkieletu litery. Została ona zaprojektowana przez autorów do identyfikacji pisanych odręcznie dużych liter alfabetu. Autorki publikacji podjęły próbę jej implementacji dla małych drukowanych liter alfabetu polskiego. Prowadzone badania wykazały, że liczba cech oryginalnej metody jest w tych warunkach niewystarczająca (alfabet polski ma 35 liter, a między niektórymi literami, np. *ą* i *g* występuje podobieństwo). Bardzo przydatna okazała się w tym przypadku implementacja wstępnej klasyfikacji liter [3].

LITERARURA

[1] Bourbakis N.O.: A parallel-symetric thinning algorithm. Pattern Recognition, Vol. 22, No. 4
[2] Brown R. M., Fay T. H., Walker C. L.: Handprinted symbol recognition system. Pattern Recognition, Vol. 21, No 2
[3] DeLuca P. G., Gisatti A.: Printed character preclassification based on word structure. Pattern Recognition, Vol. 24 No 7
[4] Dołęga C. G.: Problemy wyszukiwania znaków w rozpoznawaniu tekstu drukowanego. Informatyka nr 5, 1991
[5] Govindan V. K.: Character recognition – a review. Pattern Recognition, Vol. 19, No. 6
[6] Govindan V. K., Shivaprasad A. P.: Pattern adaptive thinning algorithm. Pattern Recognition Vol. 20, No 6
[7] Huahg J. S., Chuang K.: Heuristic approach to handwritten numeral system. Pattern Recognition Vol. 19, No. 1
[8] Kulikowski J. L.: Komputerowa analiza obiektów i rekonstrukcja relacji. Informatyka nr 7, 1991
[9] Kumar P., Bhatnager D., Rao P. S.: Pseudo one pass thinning algorithm. Pattern Recognition Letter Vol. 12, No. 9
[10] Kundu A., He Y., Bahl P.: Recognition of handprinted word: first and second hidden Markov model based approach. Pattern Recognition, Vol. 22, No 3
[11] Mantas J.: An overview of character recognition methodologies. Pattern Recognition, Vol. 19, No. 6
[12] Mantas J.: Methodology in pattern recognition and image analysis. Pattern Recognition, Vol. 20, No 1
[13] Porod W., Ferry D. K.: Pattern recognition in highly integrated circuits. Pattern Recognition, Vol. 18, No. 2
[14] Ramesh S. R.: A generalized character recognition algorithm: a graphical approach. Pattern Recognition, Vol. 22, No. 14
[15] Stefandi R.: A comment on an investigation into the skeletonization approach of hilditch. Pattern Recognition Vol. 19, No. 1
[16] Zawada A.: Automatyczne czytanie pisma. Informatyka nr 7, 1991.

## Wskazówki dla Autorów

Nadsyłane artykuły nie mogą być publikowane lub przeznaczone do opublikowania w innych czasopismach.

Materiał oprócz tekstu zasadniczego powinien zawierać:

- ★ krótki życiorys zawodowy Autora i jego zdjęcie,
- ★ wykaz literatury,
- ★ tabele,
- ★ materiał ilustracyjny (rysunki, zdjęcia czarno-białe, wydruki) dołączony do artykułu (nie wklejać materiału w tekst),
- ★ podpisy pod ilustracje.

Na osobnej stronie prosimy podać: tytuł naukowy, nazwisko i imię, nazwę zakładu pracy, numer telefonu służbowego oraz informację, jaką drogą przekazać honorarium – kasa Wydawnictwa, poczta, konto.

W związku z naliczaniem podatku dochodowego prosimy także podać:

- ★ oprócz nazwiska i imienia – także drugie imię,
- ★ imiona rodziców,
- ★ datę i miejce urodzenia,
- ★ numer identyfikacyjny PESEL,
- ★ miejsce zameldowania,
- ★ adres Urzędu Skarbowego właściwego dla miejsca zamieszkania Autora.

Tekst artykułu prosimy dostarczyć w formie maszynopisu lub wydruku komputerowego, pisanego jednostronnie, z podwójnym odstępem (30 wierszy na stronie i 60 znaków w wierszu).

Rysunki powinny być czytelne i mogą być wykonane ołówkiem.

Autor opublikowanego artykułu otrzymuje honorarium i bezpłatny egzemplarz okazowy.

Materiałów nie zamówionych redakcja nie zwraca.

TERA Spółka Z O. O.

Przedsiębiorstwo Popierania Postępu TERA Spółka z o.o. uprzejmie informuje, że posiadając kilkuletnie doświadczenie w instalacji systemów wspomagających zarządzanie

**o f e r u j e :**

- opracowanie projektów systemów,
- optymalny kosztowo i rozwojowo dobór sprzętu i oprogramowania,
- instalację systemu u klienta,
- bezpłatnie przez rok konserwację oprogramowania oraz serwis sprzętu wraz z doradztwem techniczno-eksploatacyjnym,
- szybką dostawę uzupełnień konfiguracji lub sprzętu mikrokomputerowego niezależnego od systemu, w tym mikrokomputerów ALR firmy Wearnes Technology.

**Wszelkie dodatkowe informacje uzyskają Państwo codziennie oprócz niedziel w Biurze Handlowym**

**40-025 Katowice, ul. Wojewódzka 31**
**tel. (faks): 155-26-72, teleks: 315448 tera pl**

*PAMIĘTAJ! Instalacje XENIX/NOVELL/PC MOS 386 oraz serwis to nasza specjalność – ZAPRASZAMY, ponieważ czterech lat doświadczeń nigdzie nie kupisz.*

*0/19/90*

**Ogłoszenie w INFORMATYCE**
**dotrze przede wszystkim do profesjonalistów**

MARIA JERCZYŃSKA
Instytut Rynku Wewnętrznego
i Konsumpcji
Warszawa

# Badania krajowego rynku komputerowego

Tekst niniejszy swą tematyką odbiega od treści, jakie w ostatnich latach są publikowane w „Informatyce". Nawiązuje natomiast do tradycji lat siedemdziesiątych i częściowo osiemdziesiątych, kiedy systematycznie obserwowano rozwój zastosowań komputerów w różnych dziedzinach. W warunkach systemu scentralizowanej organizacji życia było to możliwe, a dla stymulowania rozwoju komputeryzacji – także pożyteczne.

Podobne obserwacje w warunkach gospodarki rynkowej są niezwykle trudne, a wg niektórych opinii – niepotrzebne. Kogóż bowiem może teraz interesować fakt, czy i w jakim kierunku rozwija się komputeryzacja. A jednak zainteresowani istnieją i nawet można ich podzielić na dwie grupy. Pierwsza grupa, mniej ważna, to dawni badacze obserwatorzy, dziś właściwie tylko hobbyści. Starają się oni nadal poznawać rzeczywistość i badać związki przyczynowe, formułować prawidłowości, mając nadzieję, że może wnioski przydadzą się do czegoś więcej, niż tylko do wzbogacenia uogólnień teoretycznych.

Druga grupa zainteresowanych jest znacznie większa i ważniejsza z ekonomicznego punktu widzenia. To ci, którzy chcą rynek komputerowy rozwijać, rozszerzając jego podaż. Inaczej mówiąc, stawiają do dyspozycji polskim użytkownikom nowe technologie, przyspieszające rozwój gospodarczy i wzrost efektywności ekonomicznej. Przynosi im to oczywiście wymienne zyski, a jednocześnie kierunek tych działań jest w pełni zgodny z interesem gospodarki narodowej. Ludzie ci powinni znać potrzeby i zamierzenia rozwojowe w poszczególnych segmentach rynku. Być może przeprowadzają w tym celu własne sondaże rynkowe.

Artykuł niniejszy zawiera informacje o niektórych wynikach badań ankietowych, przeprowadzonych w połowie ub. r. na próbie przedsiębiorstw, głównie handlowych*). Są to informacje bardzo aktualne, odsłaniające niektóre cechy rynku komputerowego w jednej z dziedzin gospodarki. Jest prawodopodobne, iż podobne cechy występują także w innych dziedzinach i dlatego ich znajomość ma z pewnością konkretną wartość użytkową.

W objętej badaniami próbie przedsiębiorstwa państwowe stanowiły 12%, spółdzielcze 59%, a spółki z o.o. – 29%. Dziedzinami ich działalności były: obrót hurtowy, sprzedaż detaliczna, usługi, gastronomia oraz produkcja.

## Sprzęt komputerowy oraz planowane zakupy

Pierwszym wnioskiem z analizy wyników badań było stwierdzenie, że poziom komputeryzacji przedsiębiorstw wyrażony ilością zainstalowanego sprzętu był na ogół wysoki. Wnioskiem drugim – że były to głównie mikrokomputery, a wśród nich dominowały modele nie najnowsze. Aż 82% przedsiębiorstw eksploatowało sprzęt klasy IBM PC, przy czym przypadki posiadania tylko pojedynczych egzemplarzy były rzadkie. Najczęściej wyposażenie składało się z 2–3 takich komputerów. Stąd stosunkowo rzadko pracowały one w sieci – zaledwie w 25% przypadków (głównie Novell oraz OA-Link). Mikrokomputerów nowszej generacji, co najmniej klasy 386 było zaledwie 6%.

Proporcje te zmieniają się przy analizie zamiarów zakupów. Mikrokomputery klasy XT i AT interesują już tylko ok. połowę badanych jednostek, zaś jedna czwarta ma zamiar wyposażyć się w modele typu 386. Zadeklarowana skłonność do modernizacji posiadanego sprzętu jest wyraźna i rokuje stały wzrost popytu na komputery.

Ciekawsza wydaje się analiza użytkowania oprogramowania, które dotąd było znacznie rzadziej przedmiotem badań.

## Oprogramowanie użytkowe

Większość stosowanych w przedsiębiorstwach handlowych pakietów programów użytkowych miała charakter ewidencyjno-kontrolny. Najczęściej były to systemy dziedzinowe i wycinkowe, rzadko – ujmujące kompleksowo działalność przedsiębiorstwa. Fakt ten, stwierdzony podczas kilkakrotnych badań ankietowych przeprowadzonych przez IRWIK w latach siedemdziesiątych i osiemdziesiątych, został potwierdzony przez badania ubiegłoroczne. Wprawdzie poszczególne podsystemy zostały rozbudowane, co szczególnie wyraźnie widać w systemie F-K, wzbogaconym obecnie o elementy analizy ekonomicznej, ale generalnie małe systemy mają nadal charakter ewidencyjno-informacyjny. W warunkach działania na rynku konkurencyjnym, przedsiębiorstwo będzie jednak musiało uzyskać w komputerach znacznie większe wsparcie. Stąd w najbliższym czasie niezbędne będą pakiety programów użytkowych, umożliwiające realizację:

- planowania, zarówno operacyjnego jak i strategiczego,
- analizy ekonomicznej, umożliwiającej obliczanie różnych wariantów wyników ekonomicznych, określonych w planie strategicznym,
- analizy rynku, stwarzającej podstawy dla prognozowania sprzedaży,
- strategii marketingowej.

Wszystkie te problemy są wzajemnie powiązane a na Zachodzie tworzą one tzw. „business plan" każdego przedsiębiorstwa. W przeprowadzonych badaniach obserwowano skłonność przedsiębiorstw do realizowania powyższych funkcji w ramach systemu informatycznego. Zainteresowanie tymi nowymi dziedzinami wykorzystania komputerów było relatywnie duże (tabela).

Wyraźnie widoczne jest większe zainteresowanie rachunkiem ekonomicznym, marketingem oraz analizą popytu w ujęciu

*) Badania przeprowadziły: Instytut Rynku Wewnętrznego i Konsumpcji (IRWiK) w Warszawie oraz Spółdzielczy Instytut Badawczy SIB w Warszawie, które dysponują pełnymi wynikami badań.

globalnym. Mimo relatywnie małych rozmiarów próby, wobec znacznej zbieżności trendów wypowiedzi można sforumować m.in. następujące wnioski:

- Niedoceniane jest jeszcze znaczenie planowania, a zwłaszcza wyznaczania celu strategicznego, warunkującego rozwój przedsiębiorstwa. Można przypuszczać, że „bussiness plan" jeszcze przez pewien czas będzie kojarzył się tylko z Zachodem zanim okaże się, że już obecnie musi on stanowić podstawę działania przedsiębiorstw na rynku krajowym.
- Podobnie mało docenianą jest analiza popytu, chociaż już wiadomo, że klienta trzeba szukać, a nawet o niego walczyć. Jednak przedsiębiorstwa nie zdają sobie jeszcze sprawy z tego, że aby klienta zdobyć, trzeba go najpierw poznać, i że rozpoznanie to powinno przeprowadzić samo przedsiębiorstwo.

**Zainteresowanie przedsiębiorstw nowymi dziedzinami wykorzystania komputerów**

| Dziedzina wykorzystania komputerów | % zainteresowanych |
|---|---|
| Planowanie: | |
| – krótkookresowe | 53 |
| – strategiczne | 29 |
| Rachunek ekonomiczny | 68 |
| Symulowanie wyniku finansowego | 56 |
| Polityka marketingowa | 59 |
| Analiza popytu w ujęciu: | |
| – globalnym | 64 |
| – asortymentowym | 41 |
| – wg artykułów | 39 |

## Tendencje rozwojowe oprogramowania

Na ograniczone przekonanie o potrzebie modernizacji zarządzania mogą wpływać następujące czynniki:

- zbyt słaby jeszcze wpływ czynnika konkurencji rynkowej,
- nieznajomość metod rozwiązywania poszczególnych problemów,
- zaniechanie wielu statystyk państwowych powoduje konieczność samodzielnego gromadzenia informacji, co oczywiście jest czaso- i kosztochłonne,
- brak rozpoznania narzędzi technicznych, umożliwiających gromadzenie i przetwarzanie tych danych oraz otrzymywanie potrzebnych ocen.

W przypadku metod trzeba przyznać, że ich stosowanie rzeczywiście wymaga pewnej znajomości statystyki, a niekiedy matematyki i dlatego nie zawsze mogą być stosowane przez samych praktyków. Bez względu jednak na obecny charakter stosowanych programów użytkowych oraz obserwowane tendencje zakupów, rozwój gospodarki rynkowej będzie wymuszał coraz większe zainteresowanie maksymalnym wykorzystaniem komputerów dla procesów dostosowawczych przedsiębiorstw na rynku. Dlatego z dużym prawdopodobieństwem można stwierdzić, że w tej dziedzinie pojawią się olbrzymie możliwości zbytu specjalistycznego oprogramowania. Obecnie firmy softwarowe ograniczają się bardzo często do sprzedaży pakietów, które były wystarczające w warunkach gospodarki scentralizowanej. Obecnie często nie odpowiadają one już nowym wymaganiom funkcjonowania i otoczenia jednostek rynkowych i dlatego każda oferta nowych pakietów o większych walorach informacyjnych, a przy tym zapewniających stosowanie dotychczasowego sprzętu komputerowego spotka się z dużym, szybko rosnącym zaintereseowaniem.

Wydaje się, że podstawowym hamulcem rozwoju produkcji tego rodzaju oprogramowania jest niechęć producentów oprogramowania do inwestowania w opracowanie pakietów prototypowych i w ich promocję. Z drugiej strony w obecnej podaży oprogramowania zaczynają przeważać produkty przestarzałe. Wytworzyła się niewątpliwa luka na krajowym rynku oprogramowania, którą powinny wykorzystać te firmy, które będą skłonne zaryzykować inwestowanie gwarantujące odniesienie pewnego skucesu.

Z wyników omówionych badań ankietowych wynika również, że zainteresowanie sfery obrotu towarowego w Polsce usługami komputerowymi jest zbieżne z tendencjami światowymi, choć o ok. 5 lat spóźnione. Dlatego ze znacznym prawodopodobieństwem można określić już teraz kolejne trendy zakupów zarówno sprzętu, jak i oprogramowania użytkowego. Dla firm komputerowych istnieje w Polsce wiele szans i możliwości rozwoju. Aby zamieniły się one w konkretne efekty ekonomiczne, konieczne jest przeprowadzenie właściwej analizy wybranego segmentu rynku.

# Koncepcja symulacji obiektowej z wykorzystaniem Języka Turbo Pascal 5.5

**dokończenie ze s. 18**

- zdefiniowanie określonych rodzajów standarowych klas obiektów trwałych i tymczasowych, które można wykorzystywać do prefiksowania klas obiektów trwałych i tymczasowych użytkownika,
- zdefiniowanie specjalizowanego języka symulacyjnego zorientowanego obiektowo,
- implementacja języka symulacji obiektowej.

Język symulacyjny zorientowany na obiekty pozwoli znacznie uprościć programowanie symulacji przez dostarczenie definicji standardowych klas obiektów trwałych, zasobów i obiektów tymczasowych z pewnymi standardowymi atrybutami i metodami. Te deklaracje automatycznie rozszerzą lub zastąpią definicje własnych klas obiektów użytkownika dla konkretnego modelu symulacyjnego oraz pozwolą na automatyczne wygenerowanie odpowiednich egzemplarzy obiektów tych klas i wykonanie ich metod inicjujących, a więc zwolnią użytkownika od ich szczegółowego programowania. Analiza zadeklarowanych klas obiektów trwałych i rodzajów metod przez nie wykorzystywanych umożliwi automatyczne generowanie metody upływu czasu. Wszystkie te rozszerzenia będą wprowadzane podczas pretranslacji programu symulacyjnego, napisanego w symulacyjnym języku obiektowym, na obiektowy Turbo Pascal 5.5.

LITERATURA

[1] Bielecki J.: Turbo Pascal 5.5. Wersja obiektowa. WNT, Warszawa 1990
[2] Huzar Z., Koleśnik K.: A Systematic Approach to a Discrete Simulation. Proc. of Int. Conf. Simulation of System'83, Praha 1983
[3]Huzar Z., Koleśnik K.: A Systematic Approach to a Simulation Model Construction. Proc. of Int. Symp. System Analysis and Simulation 1985, Berlin 1985
[4] Koleśnik K.: A PASCALSIM Symulation Language. System Anal. Model. Simul. 5/1988, 2, s. 153–163
[5] Koleśnik K.: Koncepcja symulacji obiektowej na bazie Turbo Pascala 5.5. Centrum Obliczeniowe Politechniki Wrocławskiej, Raport P/7/1991
[6] Marciniak A.: Turbo Pascal 5.5. PWN. Warszawa – Poznań 1990
[7] Oktaba H., Ratajczak W.: SIMULA 67. WNT, Warszawa 1980
[8] Turbo Pascal Version 6.0 Programmers Guide. Borland International Inc.

## Mikroprocesory z zegarem o częstotliwości 100 MHz

Jak podano na ostatniej Międzynarodowej Konferencji Układów Półprzewodnikowych firma Intel dysponuje doświadczalnym mikroprocesorem i486 pracującym z częstotliwością zegara 100 MHz. Układ ten nie prędko pojawi się na rynku – nie ma jeszcze układów o częstotliwości 50 MHz – ale wskazuje, w jakim kierunku zmierzają prace konstrukcyjne, których celem ma być układ zawierający 100 milionów tranzystorów, pracujący z częstotliwością zegara 250 MHz i umożliwiający wykonywanie dwu miliardów rozkazów na sekundę.

Osiągnięcie firmy Intel było możliwe dzięki zastosowaniu wielu nowych rozwiązań. Układ zrealizowano w technologii CMOS o podstawowym wymiarze 0,8 $\mu$m, podczas gdy parametr ten w realizowanych dotąd układach wynosi 1 $\mu$m. Ponadto zastosowano tu projektowanie trójpoziomowe, które dodatkowo skraca połączenia między bramkami i przyspiesza działanie układu. Opracowano specjalne narzędzia programowe, które pozwalają uzyskać rozmieszczenie elementów z pełnym wykorzystaniem trzeciej warstwy. Specyficzną cechą tego rozwiązania jest rozprowadzanie zegara za pomocą synchronicznej pętli fazowej. Pozwala to zredukować wiele parametrów czasowych np. czas wyprzedzenia z 3 do 1,5 ns, czas utrzymywania z 2,5 do 1 ns, a cały czas wyjściwy z 9 do 7,5 ns.

Projekt wykazuje też kilka niedogodności. Poprawa jego działania jest zależna od mocy przychodzącego sygnału. Brak pełnego wysterowania może prowadzić do znacznych błędów systemowych. Inną istotną wadą jest brak możliwości pracy w stanie spoczynkowym, gdy układ nie pobiera energii. Przy 100 MHz moc rozpraszana w układzie wynosi 8 W, co dyskwalifikuje jego stosowanie w małych, przenośnych komputerach.

Układ został wyprodukowany i przebadany przy częstotliwościach do 100 MHz, napięciu zasilającym 5 V w temperaturze pokojowej. Przypuszcza się, że tak szybki układ będzie częścią rozwiązania wieloukładowego lub wielomodułowego, a nie typowej karty.

Przed rozpoczęciem masowej produkcji wiele problemów trzeba będzie jeszcze rozwiązać. Chodzi o zakłócenia o częstotliwościach radiowych oraz sposób chłodzenia.

Efektywność systemu komputerowego wykorzystującego ten układ będzie w znacznym stopniu zależeć od pracy szybkiej pamięci typu cache. Dlatego wzrost szybkości działania nie będzie w pełni proporcjonalny do częstotliwości zegara, ale odchylenie to nie powinno być znaczne.

Podobny mikroprocesor zapowiedziała też firma NEC, określając go jako 64-bitowy, potokowy system wektorowy. Może on wykonywać pięć operacji równolegle, co teoretycznie pozwala osiągnąć 200 mln operacji zmiennoprzecinkowych na sekundę (MFLOPS). Rozwiązanie technologiczne jest zbliżone do układu Intela – podstawowy wymiar wynosi również 0,8 $\mu$m, a rozwiązanie jest też trójpoziomowe. Układ ten może pracować z częstotliwością zegara 100 MHz.

Wydaje się, że takie rozwiązania staną się popularne w najbliższej przyszłości i coraz więcej wytwórców półprzewodników będzie produkować bardzo szybkie układy.

★ ★ ★

## Mikroporocesory 32-bitowe

Firma Adams Co. z Palo Alto w Kalifornii przeprowadziła ankietę wśród czytelników pisma Electronic Design na temat zastosowań mikroprocesorów 32-bitowych. Okazuje się, że najwięcej tych mikroprocesorów stosuje się w urządzeniach militarnych i aeronautycznych – 32,4%, a następnie w stacjach roboczych projektowania wspomaganego komputerowo CAE/CAD – 17,6%. Na następnych miejscach znajdują się systemy i sprzęt komunikacyjny oraz komputery osobiste – po 16,2%. Dalej idą sterowniki i układy obsługujące urządzenia perferyjne oraz komputery z wielodostępem (po 14,7%). Dalsze miejsca zajmują zastosowania graficzne (13,2%), systemy obsługi laboratoriów naukowych (10,3%), urządzenia i systemy pomiarowe oraz systemy sterowania produkcją (po 8,8%). Przedostatnie miejsce na liście znaczniejszych zastosowań 32-bitowych procesorów zajmuje robotyka (5,9%), a ostatnie – przemysł samochodowy (tylko 1,2%, gdyż stosuje się tu głównie prostsze mikroprocesory). Pozostałe zastosowania stanowią łącznie 8,8%. To, że suma wszystkich udziałów przekracza 100% wynika z faktu, że ankietowani podawali często więcej niż jedno zastosowanie.

Ilościowo, wzrost sprzedaży procesorów 32-bitowych kształtował się w latach 1986–1990 na poziomie ok. 86% rocznie (od 1,4 mln w 1986 r. do 15,7 mln w 1990 r.). Największy przyrost notowano w latach 1987–1988 (ponad 120%). Na lata dziewięćdziesiąte przewiduje się zmniejszenie tempa wzrostu, średnio do 26% rocznie w latach 1990–1996 (od 47% w latach 1990–1991 do 13% w latach 1995–1996). Liczba wyprodukowanych mikroprocesorów 32-bitowych ma osiągnąć w 1996 r. 61 mln sztuk.

Natomiast wartościowo, w związku z obniżeniem się ich cen, wzrost ten w latach dziewięćdziesiątych będzie nieznaczny. Pod koniec lat osiemdziesiątych, gdy wzrastała nie tylko liczba wyprodukowanych układów, ale stawały się one coraz bardziej złożone, powodując wzrost ich ceny – przyrost wartościwy w latach 1986–1988 wyprzedzał wzrost ilościowy, osiągając aż 167% rocznie. W latach 1988–1990 tempo to spadło do poziomu 25%. W 1986 r. za sprzedane mikroprocesory 32-bitowe uzyskano 114 mln dolarów, w dwa lata później już 810 mln, a w 1900 r. – 1,25 mld. W latach 1990–1996 średnie tempo wzrostu wynieść ma zaledwie 2,3 %, przy czym na lata 1993–1994 i 1995–1996 przewiduje się nawet nieznaczne zmniejszenie sprzedaży lub stagnację. Maksymalną kwotę, jaka ma być uzyskana ze sprzedaży tych mikroprocesorów w 1995 lub 1996 roku, szacuje się na niespełna 1,5 ml dolarów. Szybciej rośnie przy tym sprzedaż mikroprocesorów typu RISC (o zredukowanej liście rozkazów) aniżeli CISC (o rozszerzonej liście rozkazów), choć tych ostatnich wytwarza się nadal więcej. W latach 1989–1993 za mikroprocesorowy CISC uzyska się nieco ponad miliard dolarów rocznie, podczas gdy układów RISC w 1989 r. sprzedano zaledwie za 60 mln, a w 1993 r. wartość ta ma osiągnąć 350 mln. Szybkość działania systemów RISC gwałtownie rośnie od około 12 MIPS na początku 1991 r. do przewidywanych 100 MIPS w roku 1995.

## Światłowody w sieciach komputerowych

Wzrastająca szybkość przetwarzania (wyrażana zwykle w milionach operacji na sekundę – MIPS) jest często niweczona ograniczeniem pasma przenoszenia sieci komputerowych w sytuacji, gdy do jednej sieci jest dołączonych wiele komputerów. Firma Synarnetics Inc. z North Billerica w stanie Massachusetts, która jako pierwsza wprowadziła technikę FDDI (*Fiber Distributed Data Interface* – interfejs danych rozprowadzanych światłowodami), oferuje obecnie grupę wyrobów, które rozwiązują ten a także pokrewne problemy.

Wyroby te pozwalają na zwielokrotnienie przepustowości standardowych (10 MHz) pasm sieci Ethernet przez ich równoległe prowadzenie, a także dołączanie tych sieci do interfejsu FDDI, zapewniając w każdym węźle pasma mierzone już w setkach MHz. Kombinacja systemu głównego, wkładanych kart i odpowiedniego oprogramowania pozwala stworzyć trzy kanały FDDI o szerokości pasma 100 MHz.

System tej firmy, poprzednio o nazwie LAN System 100, a obecnie Lanplex 5000, różni się tym od rozwiązań łączących sieci takich firm, jak Cisco Systems lub Wellfleet Communication tym, że komputerowe stacje końcowe są dołączane do tej sieci. Zasadą układów firmy Synernetics jest stosowane FDDI jako podstawowej techniki łączenia rozproszonych komputerów różnych wytwórców. Firma została utworzona we wrześniu 1988 r. a w marcu 1990 r. zaczęła dostarczać oprogramowanie o nazwie FDDI *Station Management* (SMT), które ujednolica dołączanie dowolnych podzespołów do sieci lokalnej FDDI i jest zawarte we wspomnianym systemie.

Jak już wspomniano, tradycyjne sieci Ethernet, zaczynają ograniczać wydajność stacji roboczych i serwerów. Użytkownicy tych sieci nie mogą przejść bezpośrednio na komunikację światłowodową i dlatego omawiane rozwiązanie łączy te dwa środowiska. Systemy główny zawiera 12 miejsc na karty (jeden centralny procesor systemu i 11 kart dodatkowych). Karta centralnego procesora wykorzystuje układ Motoroli 68030 i obejmuje zarządzanie systemu i jego diagnostykę, ale bez przetwarzania pakietów danych. Można tu dołączać różne systemy dostępu FDDI i koncentratory, a także karty typu Express Ethernet, które można wstawiać do dowolnego złącza.

Karta dostępu FDDI łączy Lanplex 5000 z koncentratorem FDDI typu podwójnego pierścienia lub wyższego poziomu, co stwarza trzy wewnętrzne połączenia. Każda karta ma cztery porty, z których każdy można dołączyć do dowolnego z trzech pierścieni FDDI.

Każda karta Express Ethernet zapewnia osiem połączeń typu 10BaseT do standardowej wersji 2.0 sieci i wykorzystuje mikroprocesor RISC AMD 29000 o szybkości 20 MIPS umożliwiający przesyłanie 60 000 pakietów na sekundę. System może obsłużyć 256 stacji końcowych. Obejmuje on również zarządzanie siecią i oprogramowanie użytkowe. Pozwala także dołączyć port Ethernet do dowolnej stacji końcowej zapewniając pełne pasmo 100 MHz. Koszt systemu wynosi około 1500 dol. za jeden port.

★ ★ ★

## Wzrost produkcji układów ASIC oraz programowanych

W ciągu pięciu najbliższych lat wartość sprzedaży układów do specjalnych zastosowań ASIC (*Application-Specific Integrated Circuits*) oraz układów programowanych wzrośnie z 7,3 mld dolarów w roku 1990 do 12,6 mld w roku 1995, tzn. o 72,6%, przy czym najszybciej mają się rozwijać układy w technologii CMOS. Układy realizowane w pełni na zamówienie nie będą rozwijać się ze względu na coraz większe możliwości układów standardowych. Wzrost produkcji układów ASIC będzie odbywać się w tempie około 10%, a układów programowanych – około 11% rocznie. Układy EPLD *(Electrically Programmable Logic Devices* – elektryczne programowalne układy logiczne) i FPGA *(Field Programmable Gate Arrays* – programowalne u użytkownika matryce bramkowe) powinny dwukrotnie, tzn. do 6% zwiększyć swój udział w całej produkcji półprzewodników.

Jeśli chodzi o zastosowania, to najwięcej tych układów (zarówno ASIC jak i programowanych) będzie stosowanych w komputerach (wzrost w latach 1990–1995 z 3,1 do 5,5 mld dol., czyli o 78%), następnie w telekomunikacji (odpowiednio z 1,5 do 2,7 mld dol., czyli o 83%), elektronice użytkowej i samochodowej (z 1,1 do 1,8 mld dol., czyli o 70%), elektronice przemysłowej i przyrządach (z 1 do 1,6 mld dol., czyli o 60%) oraz w urządzeniach wojskowych i kosmicznych (z 0,7 do 1,1 mld dol., czyli o 57%).

Kolumny „Ze świata" oprac.

JAN RYŻKO

**W związku z wprowadzeniem powszechnego podatku dochodowego i koniecznością informowania właściwych Urzędów Skarbowych o wszystkich wypłatach zwracamy się do Autorów wysyłających teksty do opublikowania o podawanie następujących danych:**

- **nazwisko, imiona (pierwsze i drugie),**
- **imiona: ojca i matki,**
- **miejsce i data urodzenia,**
- **numer identyfikacyjny PESEL (wpisany przez Biuro Meldunkowe do dowodu osobistego – nie mylić z numerem dowodu osobistego!),**
- **dokładny adres (miejsce zameldowania),**
- **adres Urzędu Skarbowego właściwego dla miejsca zamieszkania Autora (bardzo ważne – bez tej informacji nie możemy przekazać honorarium do wypłaty!).**

**Prosimy także – do naszej wiadomości – podać numer telefonu służbowego i ustalić formę przekazania honorarium (kasa Wydawnictwa, poczta, konto).**

# W UNIX-ie ostrożność popłaca

Oczywiście UNIX, tak samo jak inne systemy, jest podatny na atak polegający na zwykłym odgadnięciu hasła użytkownika. Praktyka wskazuje, że tym zakresie największe znaczenie ma inwestycja i wysiłek włożone w sformułowanie hasła. Zaskakująco duży procent haseł można odgadnąć bezpośrednio z nazwy użytkownika oraz informacji zawartych w ogólnodostępnym pliku /etc/passwd (takich, jak imię, nazwisko, numer telefonu) uzupełnionych łatwymi do zdobycia danymi z osobistego życia ofiary ataku (imiona najbliższych, numer rejestracyjny samochodu itp.). Z kolei przymus stosowania losowo wygenerowanych haseł stanowi groźbę ujawnienia osobistych notatek, które praktycznie są niezbędne do zapamiętania nie kojarzących się z niczym wyrazów.

Bezpośrednim celem ataku na integralność systemu zabezpieczeń jest najczęściej zdobycie uprawnień administratora, ponieważ usuwa to wszelkie dalsze przeszkody. W UNIX-ie łatwy do wykonania jest atak typu „konia trojańskiego", polegający na zasymulowaniu na terminalu działania programu *login,* obsługującego weryfikację użytkownika i otwarcie sesji. Program symuluje zachętę do podania nazwy użytkownika i hasła, po czym wypisuje:

*login incorrect*

symulując błąd użytkownika przy wprowadzaniu hasła, czego nie można zweryfikować, ze względu na wyłączenie echa. Po zapisaniu skradzionego w ten sposób hasła w bezpieczynm miejscu, „koń trojański" wywołuje następnie zwykły program *login,* pozwalając niczego nie podejrzewającemu użytkownikowi podjąć następną próbę otwarcia sesji. Warto zauważyć, że tego rodzaju atak może wykonać jedynie osoba już uprawniona do korzystania z danej instalacji, gdyż wymaga to uruchomienia programu.

Powyższy przykład ilustruje ogólną metodę polegającą na skłonieniu kogoś do działania w zafałszowanym środowisku. W tym przypadku był to terminal specjalnie przygotowany jako pułapka. Podobna sytuacja jest możliwa także w trakcie sesji, jeśli zmienna Shella PATH, określająca ścieżkę poszukiwań wywoływanych zleceń, różni się od oczekiwanej. Może to spowodować, że wywołanie *su* uruchomi podobnego „konia trojańskiego". To samo dotyczy w zasadzie każdego programu, zwłaszcza gdy ofiarą ataku jest sam administrator systemu.

W UNIX-ie uprawnienia procesu zależą od tego, kto wywołał dany program. Podłożenie adminstratorowi systemu do wykonania dowolnego programu sprawi, że zawarte w nim akcje zostaną wykonane z pełnymi uprawnieniami. Doświadczony administrator oczywiście nie da się namówić wprost na uruchomienie nieznanego programu, ale może to zrobić nieświadomie. Wystarczy, jeśli w jego ścieżce poszukiwań bieżący skorowidz wystąpi wcześniej niż np. */bin* (wielu użytkowników ustawia taką właśnie kolejność, aby przyspieszyć wyszukiwanie często używanych własnych programów). Wtedy niewinne wywołanie:

*ls*

może uruchomić nie program z pliku */bin/ls,* ale *./ls.* Szczególną cechą UNIX-a jest łatwość wywołania, po wykonaniu akcji stanowiących cel ataku, prawdziwego programu *ls* tak, aby ofiara ataku niczego nie zauważyła.

Do przełamania barier ochrony wystarczy stworzenie sytuacji, w której w systemie istnieje jakikolwiek plik należący do użytkownika będącego celem ataku i mający nadane uprawnienia do wykonania i zapisu dla obcych oraz bit *set-uid.* W tej sytuacji zawartość pliku może zostać wymieniona na inny program, który zawsze będzie wykonywał się z uprawnieniami właściciela pliku. Pewne zabezpieczenie stanowi tu modyfikacja jądra systemu, wygaszająca bit *set-uid* przy każdej zmianie zawartości pliku. Nie rozwiązuje to jednak problemu, gdyż „podłożony" do wykonania program (w tym przykładzie – *ls*) może zaledwie ustawić *set-uid* w już wcześniej przygotowanym pliku.

Z poprzedniego przykładu widać, że jedną z metod ataku jest doprowadzenie do wykonania pewnej czynności w imieniu administratora systemu. Literatura zawiera wiele niecodziennych metod osiągnięcia tego celu. Wykorzystując np. fakt, że pliki specjalne */dev/tty** zwykle są dostępne do zapisu przez wszystkich, można w pewnych okolicznościach spowodować wysyłanie komunikatu, który zostanie zinterpretowany przez system tak, jak gdyby został napisany przez użytkownika na klawiaturze. Szeroko rozpowszechniony terminal VT220 po otrzymaniu znaku *ENQ* (ASCII 5) odpowiada odesłaniem komunikatu zapisanego przez użytkownika w pamięci stałej terminala. Ktoś może ukradkiem wpisać tam np.:

*passwd-d*

a później, gdy na tym stanowisku pracuje *root,* wykonać z innego terminala

*echo"\05" > /dev/console*

pozbawiając hasła administratora systemu.

Podobne zagrożenie powstaje wtedy, gdy korzysta się ze zdalnego dostępu do maszyn UNIX-owych za pomocą programu *cu.*

Podstawową trudnością, z jaką musi sobie poradzić napastnik jest tu nie tylko wykonanie ataku, ale ukrycie tego faktu. Nawet jednorazowe naruszenie integralności systemu zabezpieczeń nie powinno wystąpić, ale staje się ono naprawdę groźne wtedy, gdy pozostanie niezauważone, otwierając w przyszłości drogę do dalszych ataków w dogodnych momentach. Wejście w posiadanie zapisywalnego pliku *set-uid* przypomina kradzież klucza. Aby posłużyć się nim do dokonania włamania, trzeba dysponować określonym czasem nieświadomości strażników.

Z pewną pomocą administratorowi systemu przychodzą narzędzia służące do rejestrowania zdarzeń związanych z bezpieczeństwem systemu. Co prawda, napastnik, który zdoła

zająć w systemie uprzywilejowaną pozycję jest w stanie zatrzeć większość śladów, ale doświadczony administrator na ogół będzie mógł dostrzec skutki nienormalnych zdarzeń. W instalacjach wymagających ścisłej ochrony dobrym zwyczajem jest kierowanie komunikatów wprost na sprzężoną z konsolą systemową drukarkę. Uniemożliwi to usunięcie śladów ewentualnego włamania.

UNIX rejestruje nieudane próby otwarcia sesji konwersacyjnych, jeśli ich liczba przekroczy określony limit. Aby uruchomić ten mechanizm, należy utworzyć plik */usr/adm/loginlog*. Program *login* dopisze na końcu tego pliku każdą nieudaną próbę podania hasła.

Istnieje możliwość rejestrowania wszystkich faktów użycia *su* do zmiany identyfikaci użytkownika. Opcję tę włącza się modyfikując plik */atc/default/su*.

Administrator na własny użytek może tworzyć narzędzia weryfikujące stan plików zawierających programy uprzywilejowane */st-uid/*. Chodzi tu o wykrycie ewentualnych modyfikacji, pozwalających niepowołanym użytkownikom wykonywać operacje wymagające uprawnień admistratora, oraz o upewnienie się, że w systemie nie pojawiły się bez jego wiedzy nowe pliki *set-uid*. Do zautomatyzowania tej czynności można wykorzystać program *find* umożliwiający wyszukanie wszystkich plików *set-uid* oraz *sum*, obliczający sumę kontrolną. Pewnym problemem jest tu ukrycie zapisu prawidłowych sum kontrolnych tak, aby napastnik nie mógł go zmodyfikować i dostosować do nowej sytuacji. W praktyce sprowadza się to do zagadnienia haseł, tzn. prefiksowania hasłem zawartości każdego pliku oraz przechowania poza systemem komputerowym obliczonej dla wszystkich plików globalnej sumy kontrolnej.

Przy okazji omawiania mechanizmów rejestrowania warto przytoczyć uwagi z pracy „UNIX Operating System Security" opublikowanej przez Gramppa i Morrisa w AT&T Technical Journal (Vol. 63, No. 8, Oct. 1984), gdzie autorzy pokazują inny aspekt zagadnienia: prowadzenie rejestru np. użycia *su* stanowi jednocześnie zagrożenie ułatwiające napastnikowi wybór listy potencjalnych ofiar, gdyż to właśnie tym użytkownikom, którzy znają hasło *root*, warto przyjrzeć się w poszukiwaniu interesujących informacji. Podobnie rejestr nieudanych prób otwarcia sesji konwersacyjnej może ujawnić czyjeś hasło, gdyż często zdarza się, że przez pomyłkę użytkownicy podają hasło w odpowiedzi na pytanie o nazwę użytkownika.

Można zaryzykować ogólne stwierdzenie, że o bezpieczeństwie instalacji UNIX-owej decyduje kompetencja i opanowanie administratora systemu. Szczególnie należy strzec się pominięcia tego drugiego czynnika, gdyż większość luk w systemie zabezpieczeń powstaje wskutek pomyłek lub niedopatrzeń wynikających z pośpiechu i chęci natychmiastowej rekacji na niecodzienną sytuację.

**PIOTR CARLSON**

---

Zachęcamy do przeczytania w majowym numerze relacji Jacka Surmy z San Francisco z zimowej konferencji stowarzyszenia USENIX i międzynarodowej wystawy UNIX-owej, które odbyły się w styczniu bieżącego roku w Kalifornii.

# Komputerowe układy sterowania numerycznego

**dokończenie ze s. 12**

Stanowi to duże ułatwienie dla programisty opracowującego algorytmy procesów. Obserwując na ekranie animowany obraz przebiegu obróbki, może on skorygować ewentualne błędy programu w fazie poprzedzającej rzeczywistą jego realizację na obiekcie.

Obraz procesu na monitorze może mieć postać dwuwymiarowego widoku obszaru symulacji – użytkownik ma wówczas możliwość wyboru płaszczyzny obserwacji – lub też formę wizerunku pseudoprzestrzennego. W tym ostatnim przypadku jest możliwe obracanie obrazu pola obróbki i oglądanie go z dowolnej strony, a także powiększanie wybranych jego fragmentów (fot. 3).

Układy sterowania numerycznego najnowszej generacji mają możliwość wzajemnego ich sprzęgania, łączenia w sieci, a także przyłączania zewnętrznych komputerów sterowania nadrzędnego typu DNC *(Direct Numerical Control)*. Komunikacja między poszczególnymi układami odbywa się na zasadzie wymiany telegramów, wysyłanych w określonych sytuacjach, a zawierających dane użytkowe lub też informacje o aktualnym stanie prowadzonych operacji technologicznych. Kontrolę nad całością realizowanego procesu wytwarzania może sprawować komputer DNC nadzorujący pracę pojedynczych układów sterowania lub też całych systemów układów CNC połączonych w sieć.

Poza funkcją monitorowania stanu procesu, komputery sterowania nadrzędnego umożliwiają także zdalne sterowanie działaniem poszczególnych obiektów wchodzących w skład systemu. W niektórych przypadkach sprzężenie między układem sterowania a komputerem DNC bywa realizowane na zasadzie pełnej emulacji funkcji klawiatury i monitora układu CNC na poziomie komputera nadrzędnego.

Możliwość wzajmnego komunikowania się układów sterowania, a także wymiany informacji z innymi urządzeniami sterującymi (sterownikami PLC i komputerami) ma ogromne znaczenie we współczesnych systemach kompleksowej automatyzacji produkcji. Prace rozwojowe dotyczące tego tematu stanowią jeden z głównych nurtów działań konstruktorów i producentów układów CNC. Inne kierunki rozwoju, to stałe powiększanie szybkości działania układów sterowania oraz wzbogacanie ich możliwości funkcjonalnych (systemowych, technologicznych i obsługowych).

---

# Kapsch AG

W hotelu Marriott w końcu lutego br. austriacka firma Kapsch AG, jedno z największych przedsiębiorstw austriackich działających w telekomunkacji i przemyśle elektronicznym – firma ze stuletnią tradycją – zaprezentowała swoje wyroby. Firma specjalizuje się w produkcji nowoczesnych urządzeń telekomunikacyjnych: aparatów telefonicznych, cyfrowych central telefonicznych, radarów, pakietów oprogramowania oraz laserowej aparatury medycznej.

Na pokazie zaprezentowano Meridian 1 – cyfrowy system komunikacyjny, umożliwiający transmisję mowy, danych, tekstów i znaków. Ma on możliwść rejestrowania numerów, z którymi łączyli się abonenci, czasu trwania rozmów, ich kosztu itp. Jest produkowany przez firmę na licencji kanadyjskiego koncernu Northern Telcom, światowego potentata rynku telekomunikacyjnego. Centrale te uzyskały świadectwo homologacji w Instytucie Łączności i zgodę Ministra Łączności na pracę w polskiej sieci telekomunikacyjnej.

| | | |
|---|---|---|
| **Kowalski K., Siemiński A.: Wykorzystanie sieci komputerowych w dydaktyce**<br>INFORMATYKA 1992, nr 4, s. 1<br>Charakterystyka etapów rozwoju możliwości i korzyści zastosowania sieci komputerowych w szkołach wyższych na przykładzie doświadczeń amerykańskich. | **Kowalski K., Siemiński A.: Networks application for didactics**<br>INFORMATYKA 1992, No. 4, p. 1<br>Characteristics of development stages as well as of possibilities and advantages resulting from connection of universities with computer networks on example of US experience. | **Kowalski K., Siemiński A.: Computernetzeanwendung im Bereich der Didaktik**<br>INFORMATYKA 1992, Nr. 4, S. 1<br>Eine Charakteristik von Entwicklungsetappen sowie von Möglichkeiten und Vorteilen der Computernetzeanwendung in Hochschulen auf Beispiel der amerikanischen Erfahrungen. |
| **Przetacznik W.: Graficzne stacje robocze SPARCstation**<br>INFORMATYKA 1992, nr 4, s. 5<br>Charakterystyka sprzętowa ośmiu modeli graficznych stacji roboczych rodziny SPARCstation firmy SUN Microsytems. | **Przetacznik W.: The SPARCstation graphic workstations**<br>INFORMATYKA 1992, No. 4, p. 5<br>Herdware characteristics of eight graphic station models of the SUN Microsystems SPARCstation family. | **Przetacznik W.: Die SPARCstation-Grafikarbeitsplätze**<br>INFORMATYKA 1992, Nr. 4, S. 5<br>Eine Charakteristik von acht Grafikarbeitsplatzmodellen der SPARCstation-Familie von SUN Microsystems. |
| **Kietzman B.: Komputerowe układy sterowania numerycznego**<br>INFORMATYKA 1992, nr 4, s. 10<br>Charakterystyka współczesnych rozwiązań sprzętowych i programowych komputerowych układów sterowania numerycznego. | **Kietzman R.: CNC systems**<br>INFORMATYKA 1992, No. 4, p. 10<br>Characteristics of contemporary CNC systems hardware and software solutions. | **Kietzman B.: Die CNC-Systeme**<br>INFORMATYKA 1992, Nr. 4, S. 10<br>Eine Charakteristik von Hardware- und Softwarelösungen der zeitgenössischen CNC-Systeme. |
| **Koleśnik K.: Koncepcja symulacji obiektowej z wykorzystaniem języka Turbo Pascal 5.5**<br>INFORMATYKA 1992, nr 4, s. 14<br>Omówienie próby zastosowania programowania obiektowego do symulacji z wykorzystaniem języka Turbo Pascal 5.5 | **Koleśnik K.: Conception for object sumulation using the Turbo Pascal 5.5**<br>INFORMATYKA 1992, No. 4, p. 14<br>Application attempt of object programming for simulation using the Turbo Pascal 5.5 | **Koleśnik K.: Objektsimulation mit Anwendung von Turbo Pascal 5.5**<br>INFORMATYKA 1992, Nr. 4, S. 14<br>Ein Anwendungsversuch der Objektprogrammierung für Simulation mit Ausnutzung von Turbo Pascal 5.5 |
| **Dąbrowska I., Depowska D.: Metody automatycznego rozpoznawania znaków**<br>INFORMATYKA 1992, nr 4, s. 19<br>Charakterystyka współczesnych metod i technologii automatycznego rozpoznawnia znaków w trybie off-line. | **Dąbrowski I., Depowska D.: Methods for automatic character recognition**<br>INFORMATYKA 1992, No. 4, p. 19<br>Characteristics of contemporary methods and technologies for off-line automatic character recognition. | **Dąbrowska I., Depowska D.: Methoden für automatisierte Zeichenerkennung**<br>INFORMATYKA 1992, Nr. 4, S. 19<br>Eine Charakteristik von zeitgenössischen Methoden und Technologien der automatisierten off-line-Zeichenerkennung. |
| **Jerczyńska M.: Badania krajowego rynku komputerowego**<br>INFORMATYKA 1992, nr 4, s. 23<br>Omówienie badań ankietowych na temat wyposażenia w sprzęt komputerowy i oprogramowanie użytkowe przedsiębiorstw polskich | **Jerczyńska M.: Research of computer home market**<br>INFORMATYKA 1992, No. 4, p. 23<br>Discussion of inquiry on hardware equipment and application software in polish enterprises. | **Jerczyńska M.: Forschung des Computerinlandsmarktes**<br>INFORMATYKA 1992, Nr. 4, S. 23<br>Eine Besprechung von einer Umfrage im Bereich der Ausstattung von polnischen Unternehmungen mit Hardware und Anwendungssoftware. |


# Ogłoszenie w INFORMATYCE szansą znalezienia właściwego kontrahenta